L10: Entry 4 of 8 File: USPT Aug 17, 2004

DOCUMENT-IDENTIFIER: US 6778981 B2
TITLE: Apparatus and method for similarity searches using hyper-rectangle based multidimensional data segmentation

Abstract Text (1):
Disclosed herein is an apparatus and method for similarity searches using hyper-rectangle based multidimensional data segmentation. The similarity search apparatus has MBR generation means, first sequence pruning means, second sequence pruning means, and subsequence finding means. The MBR generation means segments a multidimensional data sequence to be partitioned into subsequences, and represents each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database. The first sequence pruning means prunes irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space. The second sequence pruning means prunes irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space. The subsequence finding means detects subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm.

Brief Summary Text (3):
The present invention relates generally to an apparatus and method for similarity searches using a hyper-rectangle based multidimensional data segmentation, and more particularly to an apparatus and method which can efficiently perform the segmentation with respect to data sets representable by multidimensional data sequences (MDS's), such as video streams, and can search for similarity using the segmentation.

Brief Summary Text (7):
As the use of multimedia data has spread to many application domains, the efficient retrieval of multidimensional, voluminous and complex information, which are the intrinsic characteristics of multimedia data, is becoming increasingly important. The present invention, as described later, belongs to retrieval technology areas for data represented by sequences, such as time-series data and multimedia data, in accordance with this retrieval requirement.

Brief Summary Text (9):
First, there is a whole sequence matching method. This method is described in detail in a thesis entitled "Efficient Similarity Search in Sequence Databases" by "R. Agrawal, C. Faloutsos, A. Swami" and published in "Proceedings of Foundations of Data Organizations and algorithms (FODO)". The method is problematic in that two sequences to be compared must be of equal length. That is, the method maps the time sequences into the frequency domain, and uses the Discrete Fourier Transform (DFT) to solve the dimensionality curse problem. In this case, each sequence whose dimensionality is reduced by using the DFT is mapped into a lower-dimensional point in the frequency domain, and is indexed and stored using R*-Tree. However, this method is limited in that a database sequence and a query sequence must be of equal length, as described above.

Brief Summary Text (10):
Second, there is a fast subsequence matching method. This method is disclosed in detail in a thesis entitled "Fast Subsequence Matching in Time-Series Databases" by "C. Faloutsos, M. Ranganathan, Y. Manolopoulos" and published in "Proceedings of ACM SIGMOD Int'l Conference on Management of Data (May, 1994.)". The basic idea of this method is that, using a sliding window with a size of w with respect to a data sequence, it represents w one-dimensional values

included in each window by a single w-dimensional point, and transforms a one-dimensional data sequence into a lower-dimensional sequence using the DFT. The lower-dimensional data sequence is partitioned into subsequences. In this case, each subsequence is represented by a Minimum Bounding Rectangle (MBR) and is indexed and stored using "ST-index". On the other hand, a query sequence is divided into one or more subsequences each with a size of w, each of which is represented by a w-dimensional point. The query processing is based on the MBRs of a data sequence stored in a database and each query point.

Brief Summary Text (12):
A query in a query process of the multidimensional sequence is given as a multidimensional sequence, and the query sequence is also divided into multiple subsequences. In one-dimensional sequence, each query subsequence is represented by a single point. However, in the multidimensional sequence, each subsequence cannot be represented by a single point, (because each point contained in each subsequence is multidimensional), such that this method cannot be used in the similarity search of the multidimensional sequence.

Brief Summary Text (17):
Meanwhile, a similarity search method for multidimensional data sequence, as proposed later in the present invention, uses a hyper-rectangle based segmentation, and technical fields related to the hyper-rectangle based segmentation are described as follows.

Brief Summary Text (28):
In accordance with one aspect of the present invention, the above object can be accomplished by the provision of an apparatus for hyper-rectangle based multidimensional data similarity searches, the multidimensional data being representable by a multidimensional data sequence, comprising MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm.

Brief Summary Text (29):
In accordance with another aspect of the present invention, there is provided an apparatus for hyper-rectangle based multidimensional data similarity searches, comprising MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance Door between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation means includes threshold calculation means for inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume and an edge using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, segment generation means for initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, geometric condition determination means for determining whether a next point of the sequence S.sub.i satisfies a geometric condition using the bounding threshold values for the volume and the edge, segment merging means for merging

the next point of the sequence S.sub.i into the current segment if geometric condition is satisfied, and segment updating means for including the current segment in the segment set and re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric condition is not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Brief Summary Text (30):
In accordance with still another aspect of the present invention, there is provided an apparatus for hyper-rectangle based multidimensional data similarity searches, comprising MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation means includes threshold calculation means for inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, segment generation means for initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, geometric and semantic condition determination means for determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume and the semantic factor, segment merging means for merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and segment updating means for including the current segment in the segment set and re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Brief Summary Text (31):
In accordance with still another aspect of the present invention, there is provided an apparatus for hyper-rectangle based multidimensional data similarity searches, comprising MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation means includes threshold calculation means for inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume, an edge and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, segment generation means for initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, geometric and semantic condition determination means for determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume, the edge and the semantic factor, segment merging means for merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and

segment updating means for including the current segment in the segment set and re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Brief Summary Text (32):
In accordance with still another aspect of the present invention, there is provided a method for a hyper-rectangle based multidimensional data similarity searches, the multidimensional data being representable by a multidimensional data sequence, comprising the steps of segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm.

Brief Summary Text (33):
In accordance with still another aspect of the present invention, there is provided a method for a hyper-rectangle based multidimensional data similarity searches, comprising the steps of segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation step includes the steps of inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume and an edge using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, determining whether a next point of the sequence S.sub.i satisfies a geometric condition using the bounding threshold values for the volume and the edge, merging the next point of the sequence S.sub.i into the current segment if the geometric condition is satisfied, and including the current segment in the segment set and updating the segment set by re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric condition is not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Brief Summary Text (34):
In accordance with still another aspect of the present invention, there is provided a method for a hyper-rectangle based multidimensional data similarity searches, comprising the steps of segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation step includes the steps of inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts and

calculating bounding threshold values for a volume and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume and the semantic factor, merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and including the current segment in the segment set and updating the segment set by re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Brief Summary Text (35):
In accordance with still another aspect of the present invention, there is provided a method for a hyper-rectangle based multidimensional data similarity searches, comprising the steps of segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation step includes the steps of inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume, an edge and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume, the edge and the semantic factor, merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and including the current segment in the segment set and updating the segment set by re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Drawing Description Text (3):
FIG. 1 is a view showing a similarity search method for a multidimensional data sequence using hyper-rectangle based segmentation from a user's and administrator's points of view according to a preferred embodiment of the present invention;

Detailed Description Text (6):
FIG. 1 is a view showing a similarity search method for a multidimensional data sequence using hyper-rectangle based segmentation from user and administrator's points of view according to a preferred embodiment of the present invention. The above similarity search method is described in detail, as follows.

Detailed Description Text (27):
On the contrary, the similarity is different from the distance. Specifically, as two objects are similar, the similarity approaches "1", while, as two objects are dissimilar, it approaches "0". The distance between two objects can be transformed into the similarity by an appropriate mapping function. Assuming that the data space is normalized in the [0,1].sup.n hyper-cube, the length of each dimension is "1" in this cube, such that the maximum allowable distance between two objects is n, the diagonal length of the cube. Accordingly, the distance between two objects can be easily mapped into the similarity. In order to measure the similarity, the Euclidean distance is used for simplicity.

Detailed Description Text (41):

In order to measure the distance between MBRs, an MBR distance D.sub.mbr between two MBRs is introduced. Generally, MBR M in the n-dimensional Euclidean space is represented by two endpoints L (low point) and H (high point) of a major diagonal of a hyper-rectangle and defined as M=(L, H). Here, L and H are defined as L=(l.sub.1, l.sub.2, . . . , l.sub.n) and H=(h.sub.1, h.sub.2, . . . , h.sub.n), respectively, where l.sub.i.ltoreq.h.sub.i (1.ltoreq.i.ltoreq.n).

Detailed Description Text (44):
The MBR distance D.sub.mbr between two MBRs, that is, A=(L.sub.A, H.sub.A) and B=(L.sub.B, H.sub.B) in the n-dimensional Euclidean space is defined as the minimum distance between two hyper-rectangles, and defined as the following Equation 4. ##EQU4##

Detailed Description Text (99):
Further, the points P.sub.j can be represented in the hyper-rectangular form for convenience by placing L.sup.i =H.sup.i =P.sup.i.sub.j for all dimensions. That is, the MBR is represented as <P.sub.j, P.sub.j, 1>. Such a rectangle is also denoted by HR(P.sub.j) which has zero volume and edge. On the other hand, the volume Vol (HR) and the total edge length Edge (HR) of the hyper-rectangle HR are defined as the following Equation [21]. ##EQU21##

Detailed Description Text (105):
Accordingly, the volume of clusters per point VPP (HR) and the edge of clusters per point EPP (HR) of the hyper-rectangle HR are calculated as the following Equation [23]. ##EQU23##

Detailed Description Text (106):
Two hyper-rectangles can be merged during the sequence segmentation process. In order to perform this merging operation, a merging operator is defined as the following Definition 8.

Detailed Description Text (108):
The merging operator .sym. between two hyper-rectangles is defined by Equation [24] below.

Detailed Description Text (109):
According to the Definition 8, it can be recognized that the merging operator .sym. has a symmetric property. That is, HR.sub.1.sym.HR.sub.2 =HR.sub.2.sym.HR.sub.1 is constructed. Consider a case that the point P is merged to the hyper-rectangle HR=<L, H, k>. This merging process will probably cause changes in the volume, the edge and the number of points of the hyper-rectangle. The amount of change in each is an important factor for clustering, and volume and edge increments can be expressed as the following Equation [25].

Detailed Description Text (116):
After the multidimensional sequences are generated from data sources, such as video clips, each sequence is partitioned into segments. The segmentation is the process for generating each segment by continuously merging a point of the sequence into a current hyper-rectangle if the predefined criteria are satisfied. Assume that a point P is merged into the hyper-rectangle HR=<L, H, k> in the unit space [0, 1].sup.n. The segmentation is done in such a way such that if the merging of the point P into the hyper-rectangle HR satisfies certain given conditions, the point P is merged into the HR of the current segment, otherwise, a new segment is started from the point P. In the segmentation process, a merging object is a hyper-rectangle, while a merged object is always a point.

Detailed Description Text (124):
Provided that a minimum hyper-rectangle containing all K points in the sequence S is HR.sub.s, a unit hyper-cube uCube is defined as a cube occupied by a single point in the space [0, 1].sup.n if all points in the sequence S are uniformly distributed over the minimum hyper-rectangle HR.sub.s. If the side-length of the cube is e, the volume and the edge are expressed as the following Equation [27]. ##EQU24##

Detailed Description Text (125):
If all points of the sequence S are uniformly scattered into the space of HR.sub.s, it can be recognized that one point is allocated to a unit hyper-cube uCube. In other words, it is intuitively seen that each point of S forms a hyper-rectangle having a unit hyper-cube shape. However, the uniform distribution assumed here is not likely to occur in reality. For example, frames in a video shot are very similar, and if each frame is represented by one point, these points are clustered together. The uniform distribution provides a geometric condition for determining whether or not the merging of two clusters is allowed.

Detailed Description Text (151):
Generation of multidimensional sequences (first step): The row materials such as videos or images are parsed to extract the feature vectors. Each vector is represented by a multidimensional point in the hyper data space. If the vector is of high dimension, a variety of dimension reduction techniques such as the Discrete Fourier Transform (DFT) or Wavelets can be applied to avoid the dimensionality curse problem. A series of points constitutes a multidimensional sequence.

Detailed Description Text (160):
On the other hand, the similarity search process proposed in the present invention can be realized by using conventional partitioning algorithms not using the sequence segmentation method proposed in the present invention. For this realization, a partitioning algorithm proposed in a thesis entitled "Fast Subsequence Matching in Time-Series Databases" by "C. Faloutos, M. Ranganathan, Y. Manolopoulos" and published in "Proceedings of ACM SIGMOD Int'l Conference on Management of Data" is used by slightly modifying its cost function.

Other Reference Publication (1):
An article entitled "Fast Subsequence Matching in Time-Series Databases," by Faloutsos et al., pp. 419-429. This article discloses a fast subsequence matching in time-series databases.

CLAIMS:

1. An apparatus for hyper-rectangle based multidimensional data similarity searches, the multidimensional data being representable by a multidimensional data sequence, comprising: MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm.

2. The similarity search apparatus according to claim 1, wherein the distance D.sub.mbr is the minimum distance between two hyper-rectangles.

4. An apparatus for hyper-rectangle based multidimensional data similarity searches, comprising: MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation means includes: threshold calculation means for inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume and an edge using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, segment generation means for initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, geometric condition determination means for determining whether a next point of the sequence S.sub.i satisfies a geometric condition using the bounding threshold values for the volume and

the edge, segment merging means for merging the next point of the sequence S.sub.i into the current segment if geometric condition is satisfied, and segment updating means for including the current segment in the segment set and re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric condition is not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

6. An apparatus for hyper-rectangle based multidimensional data similarity searches, comprising: MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation means includes: threshold calculation means for inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, segment generation means for initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, geometric and semantic condition determination means for determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume and the semantic factor, segment merging means for merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and segment updating means for including the current segment in the segment set and re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

8. An apparatus for hyper-rectangle based multidimensional data similarity searches, comprising: MBR generation means for segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; first sequence pruning means for pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; second sequence pruning means for pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned by the first sequence pruning means in a multidimensional Euclidean space; and subsequence finding means for finding subsequences similar to the given query sequence by obtaining sets of points in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation means includes: threshold calculation means for inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume, an edge and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, segment generation means for initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, geometric and semantic condition determination means for determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume, the edge and the semantic factor, segment merging means for merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and segment updating means for including the current segment in the segment set and re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points

contained in the current segment exceeds the minimum number of points per segment minPts.

10. A method for a hyper-rectangle based multidimensional data similarity searches, the multidimensional data being representable by a multidimensional data sequence, comprising the steps of: segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm.

11. The similarity search method according to claim 10, wherein the distance D.sub.mbr is the minimum distance between two hyper-rectangles.

13. A method for a hyper-rectangle based multidimensional data similarity searches, comprising the steps of: segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation step includes the steps of: inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume and an edge using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, determining whether a next point of the sequence S.sub.i satisfies a geometric condition using the bounding threshold values for the volume and the edge, merging the next point of the sequence S.sub.i into the current segment if the geometric condition is satisfied, and including the current segment in the segment set and updating the segment set by re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric condition is not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

15. A method for a hyper-rectangle based multidimensional data similarity searches, comprising the steps of: segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation step includes the steps of: inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts and calculating bounding threshold values for a volume and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the

bounding threshold values for the volume and the semantic factor, merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and including the current segment in the segment set and updating the segment set by re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

17. A method for a hyper-rectangle based multidimensional data similarity searches, comprising the steps of: segmenting a multidimensional data sequence to be partitioned into subsequences, and representing each subsequence by each Minimum Bounding Rectangle (MBR), such that sets of MBRs are generated from the multidimensional data sequence, and the MBR sets are stored in a database; pruning irrelevant data sequences using a distance D.sub.mbr between MBRs extracted from an inputted query sequence and the MBR sets stored in the database in a multidimensional Euclidean space; pruning irrelevant data sequences using a normalized distance D.sub.norm between MBRs extracted from the query sequence and the MBR sets of data sequences remaining after the data sequences are pruned in a multidimensional Euclidean space; and finding subsequences similar to the given query sequence by obtaining sets of points contained in MBRs involved in a calculation of the distance D.sub.norm from each sequence obtained using the distance D.sub.norm ; wherein the MBR generation step includes the steps of: inputting a multidimensional sequence S.sub.i and the minimum number of points per segment minPts, and calculating bounding threshold values for a volume, an edge and a semantic factor using a unit hyper-cube occupied by a single point in n-dimensional unit space, if points are uniformly distributed in a hyper-rectangle which is a minimum bounding rectangle containing all points in the sequence S.sub.i, initializing a segment set and an outlier set to empty sets and generating a current segment using a first point of the sequence S.sub.i, determining whether a next point of the sequence S.sub.i satisfies geometric and semantic conditions using the bounding threshold values for the volume, the edge and the semantic factor, merging the next point of the sequence S.sub.i into the current segment if the geometric and semantic conditions are satisfied, and including the current segment in the segment set and updating the segment set by re-generating a new current segment using the next point of the sequence S.sub.i, if the geometric and semantic conditions are not satisfied and the number of points contained in the current segment exceeds the minimum number of points per segment minPts.

Previous Doc Next Doc Go to Doc#

L17: Entry 15 of 18 File: USPT Dec 14, 1999

DOCUMENT-IDENTIFIER: US 6003029 A
TITLE: Automatic subspace clustering of high dimensional data for data mining applications

Brief Summary Text (14):
An m-dimensional data space S=A.sub.1 x A.sub.2 x . . . . x A.sub.m can be viewed as being partitioned into non-overlapping rectangular units. Each unit has the form {r.sub.1, . . . , r.sub.m }, r.sub.j =<a.sub.j, l.sub.j, u.sub.j > such that l.sub.j .ltoreq.u.sub.j, where l.sub.j, u.sub.j .epsilon. A.sub.j and a.sub.h for j.noteq.h. The units have been obtained by some partitioning into intervals (e.g., equi-width, user-specified, etc.) of each of the A.sub.i. The partitioning scheme can be different for each A.sub.i.

Brief Summary Text (18):
A region R is said to be contained in a cluster C if R.andgate.C=R. A region can be expressed as a DNF expression on intervals of the domains A.sub.i. A region R contained in a cluster C is said to be maximal if no proper superset of R is contained in C. A minimal description of a cluster is a non-redundant covering of the cluster with maximal regions. That is, a minimal description of a cluster C is a set S of maximal regions of the cluster such that their union equals C, but the union of any proper subset of S does not equal C.

Brief Summary Text (22):
The problem of covering marked boxes in a grid with rectangles has been addressed in logic minimization by, for example, S. J. Hong, MINI: A heuristic algorithm for two-level logic minimization, Selected Papers on Logic Synthesis for Integrated Circuit Design, R. Newton, editor, IEEE Press, 1987. It is also related to the problem of constructive solid geometry (CSG) formula in solid-modeling, such as disclosed by D. Zhang et al., Csd set-theoretic solid modelling and NC machining of blend surfaces, Proceedings of the Second Annual ACM Symposium on Computational Geometry, pages 314-318, 1986. These techniques have also been applied for inducing decision rules from examples, such as disclosed by S. J. Hong, R-MINI: A heuristic algorithm for generating minimal rules from examples, 3rd Pacific Rim Int'l Conference on AI, August 1994. However, MINI and R-MINI are quadratic in the size of input (number of records). Computational geometry literature also contains algorithms for covering points in two-or three-dimensions with minimum number of rectangles, for example, see D. S. Franzblau et al., An algorithm for constructing regions with rectangles: Independence and minimum generating sets for collections of intervals, Proc. of the 6th Annual Symp. on Theory of Computing, pages 268-276, Washington D.C., April 1984; R. A. Reckhow et al., Covering simple orthogonal polygon with a minimum number of orthogonally convex polygons, Proc. of the ACM 3rd Annual Computational Geometry Conference, pages 268-277, 1987; and V. Soltan et al., Minimum dissection of rectilinear polygon with arbitrary holes into rectangles, Proc. of the ACM 8th Annual Computational Geometry Conference, pages 296-302, Berlin, Germany, June 1992.

Brief Summary Text (23):
Some clustering algorithms used for image analysis also find rectangular dense regions, but they have been designed for low-dimensional datasets. For example, see M. Berger et al., An algorithm for point clustering and grid generation, IEEE Transactions on Systems, Man and Cybernetics, 21(5):1278-86, 1991; P. Schroeter et al., Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement, Pattern Recognition, 25(5):695-709, May 1995; and S. Wharton, A generalized histogram clustering for multidimensional image data, Pattern Recognition, 16(2):193-199, 1983.

Detailed Description Text (5):
For the first phase of the present invention, the simplest way for identifying dense units would be to create a histogram in all subspaces and count the points contained in each unit during one pass over the data. However, this approach is infeasible for high dimensional data.

Consequently, the present invention uses a bottom-up cluster identification approach that is similar to the conventional Apriori algorithm for finding Association rules, as disclosed by R Agrawal et al., Fast Discovery of Association Rules, Advances in Knowledge Discovery and Data Mining, U. M. Fayyad et al., editors, AAAI/MIT Press, Chapter 12, pages 307-328, 1996, and incorporated by reference herein. A similar bottom-up cluster identification approach for determining modes in high-dimensional histograms is disclosed by R. S. Chhikara et al., Register A numerical classification method for partitioning of a large multidimensional mixed data set, Technometrics, 21:531-537, 1979.

Detailed Description Text (35):
For a general set cover, the Addition Heuristic provides a cover within a factor in n of the optimum, as disclosed by L. Lovasz, On the ratio of the optimal integral and fractional covers, Discrete Mathematics, 13:383-390, 1975. Consequently, the Addition Heuristic would appear to be the preferred heuristic of the two heuristics because its quality of approximation matches the negative results of U. Feige, supra, and C. Lund et al., supra. However, implementation of an Addition Heuristic in a high-dimensional geometric setting requires a complex computation of the number of uncovered units that a candidate maximal region will cover. The residual uncovered regions that arise as the cover is formed can be complicated. Indeed, this is of interest in its own right to the approximation algorithms community, where the conventional wisdom holds that the Addition Heuristic is the preferred heuristic for set cover.

Detailed Description Text (39):
A modified version of the synthetic data generation program used by M. Zait et al., A Comparative study of clustering methods, FGCS Journal Special Issue on Data Mining, 1997, and incorporated by reference herein, was used for a comparative study of conventional clustering algorithms. The data generation program produced datasets having clusters of high density in specific subspaces and provided parameters for controlling the structure and the size of datasets, such as the number of records, the number of attributes, and the range of values for each attribute. A bounded data space (n-dimensional cube) that data points live in was assumed. Each data space was partitioned into a multi-dimensional grid generated by an equi-width partitioning of each dimension into 10 partitions. Each box of the grid formed a unit.

Other Reference Publication (8):
S. Agarwal et al., On the Computation of Multidimensional Aggregates, Proceedings of the 22nd VLDB Conference Mumbai (Bombay), India, 1996, pp. 1-16.

Other Reference Publication (9):
R. Agrawal et al., An Interval Classifier for Database Mining Applications, Proceedings of the 18th VLDB Conference, Vancouver, British Columbia, Canada, 1992, pp. 1-14.

Previous Doc Next Doc Go to Doc#

L17: Entry 8 of 18 File: PGPB Mar 27, 2003

DOCUMENT-IDENTIFIER: US 20030061213 A1
TITLE: Method for building space-splitting decision tree

Detail Description Paragraph:
[0036] A traditional decision tree makes splits based on single attribute values. The traditional decision tree studies the distribution of the multidimensional points projected into a one dimensional space, and tries to separate the points of different class labels. When this is not successful, a natural extension would be studying the distribution of the points projected into a two or even higher dimensional subspace to identify clustered regions of the different classes. That is to say the invention can now split subspaces (not just 1-dimensional intervals) in building the decision tree. While it is computationally prohibitive to search for all the clusters, it is more feasible to search for clusters formed by points in the biased target class.

Detail Description Paragraph:
[0038] The method of the invention further improves the classification by performing a nearest neighbor search after the leaf node that holds the prediction for an unknown instance is identified. The nearest neighbor search can identify the closest matching patterns for an unknown instance, but is generally less efficient than the decision tree. The method combines the two approaches to provide nearest neighbor matching with the efficiency and interpretability of the decision tree. The method introduces the notion of a scoring function and that of a nearest leaf node set for each leaf node and performs a nearest neighbor search among the nearest leaf node set based on the scoring function. The scoring function also provides an effective means for target selection.

Detail Description Paragraph:
[0059] The method only works on positive records in the data set. Since positive records only account for a small portion (such as, e.g., 1%) of the entire data set, the positive records are stored in memory. The support of a cluster is defined as n_p'/n_p, where n_p' is the number of positive records in the cluster, and n_p is the total number of positive records in the current node. The method computes minsup, i.e., the minimal support of a positive cluster that can possibly provide a gini index lower than those given by single-dimension partitions (step 401). The minsup is computed as follows: minsup=(2q-2q 2-G_best)/(2q-2q 2-qG_best), where G_best is the smallest gini index given by single attribute partitioning, q is n_p/n_n, and n_n is the total number of records in the current node.

Detail Description Paragraph:
[0064] FIG. 5 is a diagram illustrating the process of discovering one-dimensional clusters of positive points (step 402 of FIG. 4), according to an illustrative embodiment of the invention. First, each dimension is divided into N=20 equal-length bins (step 501). One scan of all the positive records is performed, and the records that fall into each bin are counted (step 502). For each dimension, steps 503-505 are performed. At step 503, a histogram is constructed for the current dimension. At step 504, the histogram for the current dimension is used to find clusters on the current dimension. Step 504 is further described below with respect to FIG. 6. At step 505, it is determined whether the current dimension is the last dimension. If so, then the method is terminated. Otherwise, the method returns to step 503 to process the next dimension.

L17: Entry 5 of 18 File: PGPB Feb 5, 2004

DOCUMENT-IDENTIFIER: US 20040024738 A1
TITLE: Multidimensional index generation apparatus, multidimensional index generation method, approximate information preparation apparatus, approximate information preparation method, and retrieval apparatus

Abstract Paragraph:
To cluster a space efficiently even in a high dimension, and realize high speed in a high dimension, and to perform similarity retrieval that can store approximate information without any waste and in a short form, can reduce an overall storage space, and can reduce the number of times of access of processing such as retrieval. There is provided a multidimensional index generation apparatus for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided area, which arranges a regular simplex to be a reference in a certain position in the multidimensional space, arranges spheres at vertexes of the arranged regular simplex, and divides the multidimensional space by the spheres.

Summary of Invention Paragraph:
[0002] The present invention relates to a retrieval apparatus that is arranged to retrieve an item similar to or identical with a designated one, and a multidimensional index generation apparatus, a multidimensional index generation method, an approximate information preparation apparatus, and an approximate information preparation method that are applied to the retrieval apparatus. In particular, the present invention relates to those apparatuses and methods that are arranged to be able to perform the retrieval and similarity retrieval at a high speed.

Summary of Invention Paragraph:
[0006] When a retrieval apparatus for performing similarity retrieval is constituted by using a computer, a plurality of characteristics (e.g., color and shape) of an object like an image (hereinafter referred to as object) are generally extracted as numerical values and are represented as points in a multidimensional space having a set of the numerical values as coordinates. If n characteristics are extracted, the characteristics are represented as points in an n-dimensional space. A dimension ranges from a few dimensions to as large as several hundred dimensions. A point corresponding to an object is correctly referred to as an object point. However, if it is not likely that misunderstanding occurs, it is simply referred to as a point.

Summary of Invention Paragraph:
[0007] A point in a multidimensional space is also considered to be a position vector from an origin. A vector is a concept of an arrow from a start point to an end point and is a concept having both a direction and a length. A start point of a vector does not have to be a specific point. However, a specific point such as an origin is considered to be a start point and a vector representing a position of the point is specifically referred to as a position vector. When one wishes to grasp a point specifically as a position vector, that is, when one wishes to grasp it as a volume having a direction and a length, a term of vector is used. In a case of an object point, a vector is referred to as an object vector or simply as a vector.

Summary of Invention Paragraph:
[0011] A simplest method of the similarity retrieval is a sequential method of checking for all points in a multidimensional space whether the points are close to a designated point. However, this method takes an extremely long time because all point records are accessed. Thus, a large number of methods are proposed which prepare an index called a multidimensional index other than the point records and use this index to reduce accesses to point records.

Summary of Invention Paragraph:

[0012] In the multidimensional index, a space is generally divided into a plurality of areas by a solid such as a cuboid or a sphere. An area occupied by this solid is referred to as a cluster. Then, points included in the cluster are managed collectively. In the SS-tree method `see [White96] D. A. White et al.: "Similarity Indexing with the SS-tree", Proc. 12.sup.th ICDE, pp.516-523 (1996)`, a sphere is a cluster and a space is divided into a plurality of spheres. In the R*tree method `see [Beckmann90] N. Beckmann: "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. SIGMOD 1990, pp.322-331 (1990)`, a cluster is a cuboid and a space is divided into a plurality of cuboids. At the time of retrieval, only clusters close to a designated point are retrieved, whereby the number of times of access to point records is reduced. In many cases, information in a cluster is accessed collectively during processing. Therefore, the information is desirably stored on a secondary memory collectively. Bringing information into this state is referred to as clustering.

Summary of Invention Paragraph:
[0013] Information on a cluster or information on points included in the cluster is managed within a multidimensional index as a retrieval record (index record). Although the index record may be referred to as an index record, it is referred to as a index record for simplicity. Any multidimensional index has this index record inside it.

Summary of Invention Paragraph:
[0020] Incidentally, various methods have been proposed conventionally concerning the multidimensional index `see [Gaede98] V. Gaede et al.: "Multidimensional Access Methods", ACM Computing Surveys, Vol. 30, No. 2, (June 1998)`. These methods are roughly classified as follows:

Summary of Invention Paragraph:
[0028] An index has a hierarchical structure. By dividing a multidimensional space into partial areas hierarchically, a retrieval range is limited to realize speed-up.

Summary of Invention Paragraph:
[0046] Incidentally, in the multidimensional index, it is important to reduce access to a point record or an index record. As a method for this, a method of reducing the number of times of access is proposed which extracts shorter information from the point record or the index record (this information is referred to as approximate information), and uses the approximate information to determine whether it is necessary to access the point record or the index record. Reducing the number of times of access using this method is referred to as filtering. Finding an approximate position is referred to as approximation. If this is compared to a map, it corresponds to finding information indicating an approximate position such as a country, a prefecture or a city as opposed to an address as accurate as a number of street.

Summary of Invention Paragraph:
[0053] Here, it is assumed that an approximate object point according to rectangular coordinates exists in a cuboid. When this cuboid is divided at an equal interval for each coordinate axis, the cuboid can be divided into a plurality of partial cuboids. This partial cuboid is referred to as a cell. Then, information on which cell an object point belongs to is assumed to be approximate information. Compared with representing an object point with accurate coordinates, since it is not seen where in the cell the object point exists, the information is approximate. However, it can be represented with far less volume for that as information.

Summary of Invention Paragraph:
[0056] (Multidimensional Index)

Summary of Invention Paragraph:
[0057] As the Internet and input apparatuses (scanner, digital camera) widespread, both the number and a volume of multimedia data are sharply increasing. As the number of multimedia data increases, a technique for retrieving the data is naturally required. In particular, in the case of multimedia, there are high expectations for similarity retrieval based on its contents. In addition, since the number of retrieval objects increases, a high-speed retrieval is required. In research and development of a multidimensional index, importance is often attached to this speed-up. A performance of the similarity retrieval is significantly affected by the number of times of input/output, and it is a key point how to reduce this number of times of input/output.

Summary of Invention Paragraph:
[0058] If the number of times of input/output is reduced, two points concerning a space efficiency and adaptability in a high dimension are important. As to the space efficiency, it is important to make a cluster and approximate information on the multidimensional index as compact as possible and reduce the number of times of input/output. As to the adaptability in a high dimension, accuracy of the similarity retrieval can be generally attained by increasing the number of characteristic volumes, that is, making a dimension of the multidimensional space higher. However, when a dimension is increased to as high as several tens dimensions to several hundred dimensions, as introduced in [Katayamaol] `see Katayama Norio et al.: "Index Technique for Similarity Retrieval", Joho shori (Information Processing) Vol. 42, No. 10, pp. 958-964, (October 2001)`, a phenomenon called a curse of dimensionality occurs, and a performance of similarity retrieval generally falls. According to the curse of dimensionality, it is known that problems such as the similarity retrieval and multivariate analysis become difficult in a high dimension. These problems are collectively referred to as the curse of dimensionality. As a concrete example, when points are uniformly distributed in a multidimensional space, a phenomenon that, in view of a certain point, other points gather near a spherical surface with the point as a center. That is, there is little difference of distances.

Summary of Invention Paragraph:
[0061] In the conventional technique, the inside of a cuboid is approximated by rectangular coordinates. On the other hand, there are provided a large number of multidimensional indexes using a sphere (see [Katayama97] `N. Katayama el al.: "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries", Proc. SIGMOD 1997, pp.369-380 (1997)`, [White96]). If it is attempted to approximate points in a sphere by a method according to a cuboid, the inside of a cube circumscribing the sphere is represented by rectangular coordinates as shown in FIG. 37. For simplicity, the case of a two-dimension will be described first. If it is attempted to approximate a point with the conventional method in the two-dimension, a result shown in FIG. 38 is obtained. Here, a point is approximated with total 256 square cells that are divided equally vertically and horizontally into sixteen pieces. A cell including a point P can be represented as (5,3). As the vertical and horizontal parts are divided into sixteen pieces, each can be represented as a bit, and total can be represented as 8 bit. However, in this case, areas such as (1,1) and (2,0) are outside the sphere. There are 40 or more such partial squares in total. That is, waste occurs in representation. This waste is less in the case of two-dimension. Next, a case of a high dimension will be described.

Summary of Invention Paragraph:
[0065] (Multidimensional Index and Filtering)

Summary of Invention Paragraph:
[0066] A database system, in particular, a relational database is becoming complicated according to expansion of a specification of SQL. As stated in [Chaudhuri00] `see S. Chaudhuri et al.: "Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System", Proc. of Intl. Conf. of Very Large Database Systems, (2000)`, since functions of a database system are expanded and optimization is complicated, maintenance, management, performance estimate and the like are becoming difficult and maintenance costs and management costs are increasing. Thus, simplification is desired. In a page based method of controlling a page that is a unit of input/output by oneself, although clustering is easily controlled, a kernel part of the database system should be manipulated. The database system is becoming huge and complicated, and a lot of studies of an expansion database for facilitating such expansion of functions are performed. However, in an actual development side, if such expansion is performed, a large amount of costs are incurred including those for tests and maintenance as an actual situation. This seems to be a reason why a method of multidimensional index is not put into practice in spite of the fact that many methods of multidimensional index are proposed.

Summary of Invention Paragraph:
[0068] Similarly, if a method for a multidimensional index can be realized on a database system, it becomes easy to put the method into practice. If it is prepared based on the standard such as SQL, it also becomes possible to run it on many existing database systems. In this case, since no manipulation can be applied to a page, the application is realized by record manipulation, that is, the application is based on a record. Although in the record based application the application is easy to realize, since clustering cannot be controlled generally, it is required to reduce the number of times of access to a record.

Summary of Invention Paragraph:
[0069] The present invention has been achieved in order to solve the above-mentioned problems, and it is an object of the present invention to provide a multidimensional index generation apparatus, a multidimensional index generation method, an approximate information preparation apparatus, an approximate information preparation method and a retrieval apparatus that can divide a sphere efficiently, can realize efficient use of a storage space, can attain speed-up of retrieval processing, and can establish the inside of a sphere with shorter approximate information to realize efficiency of a storage space and cost reduction, thereby being able to easily perform establishment of a system.

Summary of Invention Paragraph:
[0070] In order to solve the above-mentioned problem, the present invention provides a multidimensional index generation apparatus for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas in order to specify a predetermined point in the multidimensional space, which includes reference regular simplex arrangement means for arranging a regular simplex to be a reference in a certain position in the multidimensional space, and sphere arrangement means for arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arrangement means and dividing the multidimensional space by the sphere. In an embodiment of the present invention, the reference regular simplex arrangement means and the sphere arrangement means are constituted by cooperation of a control device 11, a sphere generation device 12 and a point generation device 13.

Summary of Invention Paragraph:
[0071] In addition, the multidimensional index generation apparatus of the present invention further includes connection regular simplex arrangement means for arranging a plurality of regular simplexes by connecting the regular simplex to another regular simplex with the same size as the regular simplex once or more such that surfaces of both the regular simplexes join each other, and the sphere arrangement means is characterized by dividing the multidimensional space by arranging a sphere on a vertex of the regular simplex arranged by the reference regular simplex arrangement means as well as vertexes of the plurality of regular simplexes arranged by the connection regular simplex arrangement means.

Summary of Invention Paragraph:
[0072] Further, in the multidimensional index generation apparatus of the present invention, the reference regular simplex arrangement means or the connection regular simplex arrangement means is characterized by arranging a further regular simplex for a sphere arranged by the sphere arrangement means and dividing the sphere in a hierarchical manner by the sphere arrangement means arranging a further sphere at a vertex of the further regular simplex.

Summary of Invention Paragraph:
[0073] In the multidimensional index generation apparatus of the present invention, the multidimensional space is a sphere as a partial space, and the reference regular simplex arrangement means may also be characterized by arranging the regular simplex to be a reference such that the center of gravity of the regular simplex to be a reference coincides with a center of the sphere.

Summary of Invention Paragraph:
[0074] In addition, in the multidimensional index generation apparatus, the multidimensional space is a sphere as a partial space, and the reference regular simplex arrangement means may also be characterized by arranging the regular simplex to be a reference such that the center of gravity of the regular simplex to be a reference coincides with a center of a substantial sphere by a point included in the sphere of the multidimensional space.

Summary of Invention Paragraph:
[0075] Moreover, the multidimensional index generation apparatus may also be characterized by including judging means for judging the number of vectors included in a sphere and vector holding means for, based on a result of judgment by the judging means, if the number of vectors included in the sphere is small, holding the vectors as they are without turning the vectors into a sphere. Note that this vector holding means is also constituted by cooperation of the control device 11, the sphere generation device 12 and the point generation device 13.

Summary of Invention Paragraph:

[0076] Moreover, the multidimensional index generation apparatus may also be characterized by including clustering means for performing clustering by arranging identifiers specifying the object point in hierarchy based on the divided sphere.

Summary of Invention Paragraph:
[0077] In addition, the present invention provides a multidimensional index generation method of dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided area, which includes a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in the multidimensional space and a sphere arrangement step of arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arrangement step and dividing the multidimensional space by the sphere.

Summary of Invention Paragraph:
[0079] In addition, the present invention provides an approximate information preparation apparatus for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to the positional information concerning the registered point in the multidimensional space, which includes vector setting means for setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, axial length calculating means for finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length, distance calculating means for finding a length from the point to the closest point on the direction vector as a distance, and approximate information forming means for forming the approximate information based on a predetermined direction vector set by the vector setting means, an axial length calculated by the axial length calculating means and a distance calculated by the distance calculating means. Further, the approximate information preparation apparatus corresponds to the approximate information generation device in the embodiment of the present invention, and the axial length calculating means, the distance calculating means and the approximate information forming means are constituted by cooperation of an arithmetic unit such as a CPU and software.

Summary of Invention Paragraph:
[0085] Moreover, in the approximate information preparation apparatus of the present invention, the vector setting means may be characterized by setting the direction vector based on each coordinate value in the case in which a predetermined point in the multidimensional space is represented by rectangular coordinates and, at the same time, setting the predetermined direction vector.

Summary of Invention Paragraph:
[0086] In addition, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by arranging a regular simplex in the multidimensional space, and using vertex vectors as a vector from the center of gravity of the regular simplex to a vertex of all or at least a part of the regular simplex to set the direction vector and, at the same time, setting the predetermined vector.

Summary of Invention Paragraph:
[0089] Moreover, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by including means for arranging a regular simplex in the multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from the center of gravity of the regular simplex to the vertex of the regular simplex, and finding vectors g(1), g(2), . . . , g(k) as

Summary of Invention Paragraph:
[0091] Moreover, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by including means for arranging a regular simplex in the multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from the center of gravity of the regular simplex to the vertex of the regular simplex,

and finding vectors g(1), g(2), , g(k) as

Summary of Invention Paragraph:
[0103] In addition, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by setting a direction vector by recursively dividing a dimension of a vector obtained by normalizing an object vector as a vector representing the predetermined point, constituting an identifier using a ratio of length, and assigning bits such that a surface area of a divided sphere and the number according to a bit assigned to a divided vector are proportional to each other.

Summary of Invention Paragraph:
[0104] In addition, the present invention provides an approximate information preparation method of, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to the positional information concerning the registered point in the multidimensional space, which includes a vector setting step of setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, a step of finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length and finding a length from the point to the closest point on the direction vector as a distance, and an approximate information forming step of forming the approximate information based on a predetermined direction vector set by the vector setting step, a calculated axial length and a calculated distance calculated by the step of finding an axial length and a distance.

Summary of Invention Paragraph:
[0106] In addition, the present invention provides a retrieval apparatus that retrieves an item identical with or similar to a designated one from a memory unit storing a plurality of objects, which includes a multidimensional index generation unit for dividing a multidimensional space into a plurality of areas to generate a multidimensional index in association with the divided areas in order to specify a predetermined object in the multidimensional space, the multidimensional index generation unit including reference regular simplex arranging means for arranging a regular simplex to be a reference in a certain position in the multidimensional space and sphere arranging means for arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arranging means and dividing the multidimensional space by the sphere, and a retrieval unit for using a multidimensional index generated by the multidimensional index generation unit to retrieve the object.

Summary of Invention Paragraph:
[0107] In addition, in the retrieval apparatus of the present invention, the multidimensional index generation unit is characterized by including an approximate information preparation unit for, in retrieving a predetermined point in a multidimensional space that is registered as a position in the multidimensional space, preparing approximate information that is obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to positional information concerning the registered point in the multidimensional space.

Summary of Invention Paragraph:
[0108] Moreover, in the retrieval apparatus of the present invention, the approximate information preparation unit may be characterized by including vector setting means for setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, axial length calculating means for finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length, distance calculating means for finding a length from the point to the closest point on the direction vector as a distance, and approximate information forming means for forming the approximate information based on a predetermined direction vector set by the vector setting means, an axial length calculated by the axial length calculating means and a distance calculated by the distance calculating means.

Summary of Invention Paragraph:
[0110] Further, in this embodiment, there is disclosed a multidimensional index generation program for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas, which is stored in a computer readable storage medium, the multidimensional index generation program causing a computer to execute a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in the multidimensional space and a sphere arrangement step of arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arrangement step and dividing the multidimensional space by the sphere.

Summary of Invention Paragraph:
[0111] In addition, in this embodiment, there is disclosed an approximate information preparation program for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to the positional information concerning the registered point in the multidimensional space, which is stored in a computer readable storage medium, the approximate information preparation program causing a computer to execute a vector setting step of setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, a step of finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length and finding a length from the point to the closest point on the direction vector as a distance, and an approximate information forming step of forming the approximate information based on a predetermined direction vector set by the vector setting step, a calculated axial length and a calculated distance calculated by the step of finding an axial length and a distance. In this case, the computer readable medium includes portable storage media such as a CD-ROM, a flexible disk, a DVD disk, a magneto-optical disk and an IC card, a database for holding a computer program, or other computers and a database therefor, and a transmission medium on a line.

Brief Description of Drawings Paragraph:
[0137] FIG. 19 is a flow chart showing a flow at the time of generation of a multidimensional index;

Detail Description Paragraph:
[0162] The similarity retrieval apparatus of the embodiment consists of a generation apparatus 1 for performing generation and update of a multidimensional index, a retrieval apparatus (similarity retrieval apparatus) 2 for using a generated multidimensional index to perform similarity retrieval and, at the same time, using approximate information to perform filtering processing, and a database 3.

Detail Description Paragraph:
[0165] The database 3 is constituted by a sphere relation database 31 for storing a sphere relation and a point relation database 32 for storing a point relation. Preparation (establishment) of a multidimensional index, preparation of approximate information and similarity retrieval (retrieval), all of which are executed in this apparatus, will be hereinafter described.

Detail Description Paragraph:
[0166] I. Establishment of a Multidimensional Index

Detail Description Paragraph:
[0167] In a multidimensional index method, first, how to constitute a multidimensional index is important. It will be hereinafter described how a multidimensional index is established by a generation apparatus.

Detail Description Paragraph:
[0184] Here, a vector from the center of gravity of the regular simplex in question to a point is referred to as a center of gravity to point vector. In addition, a surface vector with the smallest angle with respect to the center of gravity to point vector among n+1 surface vectors of the regular simplex is found, and a new regular simplex is connected to a surface

corresponding to this surface vector such that surfaces match each other well. In this way, new regular simplexes are created one after another. This is called growth of a regular simplex. Every time a regular simplex grows, a newly generated regular simplex approaches the point. In the basic division, the number of spheres is n+1 at the maximum, and a size of a sphere is generally large. A radius of a sphere is restricted by a distribution of a set of points.

Detail Description Paragraph:
[0185] Next, a method of dividing a space by a sphere of an arbitrary radius will be described more generally. FIG. 4 is a diagram showing the case of the two-dimension. In this figure, as it is well known, a two-dimensional space (plane) is covered by circles of the same radius without a gap and with least overlapping. FIG. 5 is a diagram in which a part of FIG. 4 is extracted and centers of the circles are connected by lines. When observed well, in this figure, regular triangles are regularly arranged and circles are arranged at vertexes thereof. This arrangement of regular triangles is attained by placing one regular triangle to be a reference first and subsequently attaching regular triangles such that sides of the regular triangles match each other.

Detail Description Paragraph:
[0186] In the case of the three-dimension, this cannot be attained so simply as in the case of the two-dimension. This is because, if a regular tetrahedron of the same size is attached to one regular tetrahedron to be a reference as in the case of the two-dimension such that surfaces match each other, a gap is generated between the regular tetrahedrons. It is known that, if five regular tetrahedrons are connected so as to draw a circle, a gap of approximately 10 degrees is generated between a first regular tetrahedron and a last regular tetrahedron. If it is attempted to draw more circles, the circles do not completely match the regular tetrahedron to be a reference but intersect it. That is, the three-dimensional space cannot be covered by regular tetrahedrons without overlapping as in the case of the two-dimension. In the three-dimension, the question of what is an arrangement without a gap and with least overlapping has been unsolved for nearly 400 years. Recently, it seems that a most densely filled structure (usually, a method adopted in filling balls in a box) has been proved to be most suitable.

Detail Description Paragraph:
[0195] <5> A surface vector with the smallest angle with respect to the center of gravity to point vector among n+1 surface vectors of the regular simplex is found, and a new regular simplex is connected to a surface corresponding to this surface vector such that surfaces match each other well.

Detail Description Paragraph:
[0234] <1> The center of gravity of a reference regular simplex a is matched with the center of the sphere Sd. A method for obtaining a radius of the reference regular simplex .sigma. is described below. A set of points included in the sphere Sd is assumed to be as follows:

Detail Description Paragraph:
[0258] <1> The center of gravity of a reference regular simplex .sigma. is matched with the center of the sphere Sd. A radius of the reference regular simplex .sigma. is made equal to the radius of the sphere Sd or larger than a radius of a root sphere. A set of points included in the sphere Sd is assumed to be as follows:

Detail Description Paragraph:
[0267] 3.3) A Method of Matching the Center of gravity of Data with the Center of Gravity of a Reference Regular Simplex

Detail Description Paragraph:
[0268] The present invention has been described on the premise that the center of gravity of the reference regular simplex is matched with a center of a parent sphere for ease of description. However, points included in the parent sphere are not always distributed around the center of the parent sphere. The points are likely to gather in a specific part of the parent sphere. In this case, the parent sphere is likely to be divided into a small number of spheres (one child sphere in the worst case). Thus, a method is possible which sets the center of gravity of a set of points included in the parent sphere, that is, a center of a substantial sphere as a center of the reference regular simplex. In this case, with the above-mentioned method of the basic division, a situation may occur in which a point is not included (left out)

in any child sphere. This is because a parent sphere is divided without any gap by the basic division only when a center of the sphere and the center of gravity of a reference regular simplex coincide with each other. Therefore, in this method, the extended division is always used. Note that it is possible to use the basic division unnaturally by enlarging a radius of a reference regular simplex. However, this is unadvisable because a radius of a child sphere is larger than a parent sphere in this case.

Detail Description Paragraph:
[0296] The storage structure of a relation has been described. Here, it will be described how an index is generated as a whole using the storage structure. FIG. 19 shows an entire flow diagram of operations of a multidimensional index generation apparatus.

Detail Description Paragraph:
[0301] The preparation of a multidimensional index by spheres has been described. More speed-up can be realized by further adding filtering by approximation to this method. This method of approximation will be described first.

Detail Description Paragraph:
[0304] A situation in which points are distributed in a sphere with a certain point as a center will be hereinafter considered. The center may be an arbitrary point but is assumed to coincide with an origin of a multidimensional space in order to simplify descriptions. This sphere is referred to as an object sphere in that object points are distributed in its inside. A radius of the object sphere may be arbitrary but is assumed to be 1 without losing generality in order to simplify descriptions as well. A sphere with a radius 1 is also referred to as a unit sphere.

Detail Description Paragraph:
[0344] A regular simplex will be considered. The center of gravity of this regular simplex is matched with an origin of an object sphere. A length to each vertex of the regular simplex from the center of gravity is assumed to be 1. Therefore, this regular simplex internally contacts the object sphere (because a radius of the object sphere is assumed to be 1 without losing generality). A vector from this center of gravity to each vertex is referred to as a vertex vector, and this vertex vector is assumed to be a direction vector. Therefore, first, n+1 direction vectors equivalent to the number of vertexes are created. These vectors are assumed to be as follows:

Detail Description Paragraph:
[0496] The method of approximating an object point has been described. However, in general, this method can approximate not only an object point but also a point. In particular, since a center of a sphere is also a point, it has a significant meaning to approximate a center. This is because, if a center can be approximated, the sphere itself can also be approximated by adding information on a radius to it. It is described above that the method of using a sphere as a cluster in a multidimensional index is used. With these methods, it is judged whether or not a sphere that is a cluster intersects vicinity and, if not intersecting, the number of times of access to a point vector or an index vector is reduced utilizing the advantage that the inside of the sphere may not be checked. If it is checked whether or not the sphere intersects the vicinity, an index record corresponding to the sphere is accessed to make the judgment from information on coordinates and a radius of its center. That is, a vector corresponding to the sphere must be accessed. However, when the sphere has been approximated, if it is found that the sphere does not intersect the vicinity from the approximate information without accessing an index record, it becomes unnecessary to access the index record.

Detail Description Paragraph:
[0502] Next, application of the above-mentioned approximation to the multidimensional index of the present invention will be described.

Detail Description Paragraph:
[0510] (b) Method Applied to a Multidimensional Index Using a Sphere

Detail Description Paragraph:
[0511] As described before, several multidimensional indexes using a sphere have been proposed. More specifically, a multidimensional index is used in SS-tree and SR-tree, and in A-tree partially. SS-tree is a first method using a sphere and is known as a high speed method.

Moreover, faster methods such as SR-tree and A-tree have been proposed which are improvements of SS-tree. In the multidimensional index using a sphere, an object point is divided by a plurality of spheres including the object point and it is judged whether or not the sphere intersects vicinity, whereby, if the sphere does not intersect the vicinity, reduction of the number of times of access to a point record is realized by utilizing the fact that it is unnecessary to check an object point included in the sphere. A set of direction vectors is decided with respect to an object point in the sphere considering that a center of the sphere is a center of the object sphere, whereby the method of the present invention can be applied. In addition, with respect to a sphere, approximate information on a sphere according to the present invention is also stored in an index record of a corresponding sphere and filtered at the time of retrieval, whereby it becomes possible to reduce the number of times of access to the index record corresponding to the sphere.

Detail Description Paragraph:
[0543] Since approximate information is used only for representing points in a sphere, points in the sphere can be approximated with less approximate information. Therefore, by applying the present invention to a multidimensional index or the like using a sphere, it becomes possible to realize similarity retrieval with less costs for reading approximate information.

Detail Description Paragraph:
[0548] As described above in detail, according to the present invention, there is an effect that a multidimensional index generation apparatus, a multidimensional index generation method, an approximate information preparation apparatus, an approximate information preparation method and a retrieval apparatus can be provided, which can divide a sphere efficiently, can realize efficiency of a storage space, can attain high speed of retrieval processing, and can establish the inside of a sphere with short approximate information to realize efficiency of a storage space and realize cost reduction, thereby performing similarity retrieval at a high speed and, at the same time, establishing apparatuses at low costs and easily.

CLAIMS:

1. A multidimensional index generation apparatus for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas in order to specify a predetermined point in the multidimensional space, comprising: reference regular simplex arrangement portion adapted to arrange a regular simplex to be a reference in a certain position in said multidimensional space; and sphere arrangement portion adapted to arrange a sphere at a vertex of said regular simplex arranged by said reference regular simplex arrangement portion and dividing said multidimensional space by said sphere.

2. The multidimensional index generation apparatus according to claim 1, further comprising connection regular simplex arrangement portion adapted to arrange a plurality of regular simplexes by connecting said regular simplex to another regular simplex with the same size as said regular simplex once or more such that surfaces of both the regular simplexes join each other, wherein said sphere arrangement portion divides said multidimensional space by arranging a sphere on a vertex of said regular simplex arranged by said reference regular simplex arrangement portion as well as vertexes of said plurality of regular simplexes arranged by said connection regular simplex arrangement portion.

3. The multidimensional index generation apparatus according to claim 2, wherein said reference regular simplex arrangement portion or said connection regular simplex arrangement portion arranges a further regular simplex for a sphere arranged by said sphere arrangement portion and divides said sphere in a hierarchical manner by said sphere arrangement portion arranging a further sphere at a vertex of the further regular simplex.

4. The multidimensional index generation apparatus according to claim 1, wherein said multidimensional space is a sphere as a partial space, and said reference regular simplex arrangement portion arranges said regular simplex to be a reference such that the center of gravity of said regular simplex to be a reference coincides with a center of said sphere.

5. The multidimensional index generation apparatus according to claim 1, wherein said multidimensional space is a sphere as a partial space, and said reference regular simplex arrangement portion arranges said regular simplex to be a reference such that the center of gravity of said regular simplex to be a reference coincides with a center of a substantial

sphere by a point included in said sphere of said multidimensional space.

6. The multidimensional index generation apparatus according to claim 1, further comprising: judging portion adapted to judge the number of vectors included in a sphere; and vector holding portion adapted to, based on a result of judgment by said judging means, if the number of vectors included in said sphere is small, hold the vectors as they are without turning the vectors into a sphere.

7. The multidimensional index generation apparatus according to claim 1, further comprising clustering portion adapted to perform clustering by arranging identifiers specifying said object point in hierarchy based on said divided sphere.

8. A multidimensional index generation method of dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided area, comprising: a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in said multidimensional space; and a sphere arrangement step of arranging a sphere at a vertex of said regular simplex arranged by said reference regular simplex arrangement step and dividing said multidimensional space by said sphere.

9. An approximate information preparation apparatus for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to the positional information concerning said registered point in said multidimensional space, comprising: vector setting portion adapted to set a set of direction vectors representing a direction in said multidimensional space and, at the same time, to set a predetermined direction vector corresponding to said predetermined point using at least a part of the set of direction vectors; axial length calculating portion adapted to find a length from an origin of said set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length; distance calculating portion adapted to find a length from the point to the closest point on the direction vector as a distance; and approximate information forming portion adapted to form said approximate information based on a predetermined direction vector set by said vector setting portion, an axial length calculated by said axial length calculating portion and a distance calculated by said distance calculating portion.

15. The approximate information preparation apparatus according to claim 9, wherein said vector setting portion sets said direction vector based on each coordinate value in the case in which a predetermined point in said multidimensional space is represented by rectangular coordinates and, at the same time, sets said predetermined direction vector.

16. The approximate information preparation apparatus according to claim 9, wherein said vector setting portion arranges a regular simplex in said multidimensional space, and uses vertex vectors as a vector from the center of gravity of said regular simplex to a vertex of all or at least a part of said regular simplex to set said direction vector and, at the same time, sets the predetermined vector.

19. The approximate information preparation apparatus according to claim 16, wherein said vector setting portion comprises: a portion for arranging a regular simplex in said multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from the center of gravity of said regular simplex to the vertex of the regular simplex, and finding vectors g(1), g(2), . . . , g(k) as g(1)=v(i(1)) g(2)=(v(i(1))+v(i(2))/2 . . . g(k)=(v(i(1))+v(i(2))+ . . . +v(i(k)))/k; a portion for finding a vector g=n((g(1)+g(2)+ . . . +g(k))/k) that is found by normalizing vectors to centers of gravity of g(1), g(2), . . . , g(k) to set them as direction vectors; and a portion for using numbers i(1), i(2), . . . , i(k) of vertex vectors as said predetermined vector to set said predetermined vector.

20. The approximate information preparation apparatus according to claim 16, wherein said vector setting portion comprises: a portion for arranging a regular simplex in said multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from

the center of gravity of the regular simplex to the vertex of the regular simplex, and finding vectors g(1), g(2), . . . , g(k) as g(1)=n(v(i(1))) g(2)=n((v(i(1))+v(i(2))/2) . . . g(k)=n((v(i(1))+v(i(2))+ . . . +v(i(k)))/k); and a portion for, based on g(1), g(2), . . . , g(k), finding a vector g(i) having a smallest argument with an object vector among them, finding a vector m(j) from the origin O to a midpoint of g(j) (j.noteq.i) and g(i) as m(j)=(g(i)+g(j))/2, finding a vector group g(1), g(2), . . . , g(k) found by normalizing this m(j), and repeating this processing t times and, thereafter, setting a direction vector by finding the center of gravity g of g(1), g(2), . . . , g(k) and normalizing the center of gravity g, and setting said predetermined vector by a set of (j1, j2, . . . , jt).

24. The approximate information preparation apparatus according to claim 9, wherein said vector setting portion sets a direction vector by recursively dividing a dimension of a vector obtained by normalizing an object vector as a vector representing said predetermined point, constituting an identifier using a ratio of length, and assigning bits such that a surface area of a divided sphere and the number according to a bit assigned to a divided vector are proportional to each other.

25. An approximate information preparation method of, in retrieving a predetermined point in a multidimensional space registered as a position in said multidimensional space, preparing approximate information obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to said positional information concerning said registered point in said multidimensional space, comprising: a vector setting step of setting a set of direction vectors representing a direction in said multidimensional space and, at the same time, setting a predetermined direction vector corresponding to said predetermined point using at least a part of said set of direction vectors, a step of finding a length from an origin of said set predetermined direction vector to a closest point from the point on said predetermined direction vector as an axial length and finding a length from the point to the closest point on said direction vector as a distance; and an approximate information forming step of forming said approximate information based on a predetermined direction vector set by said vector setting step, a calculated axial length and a calculated distance calculated by said step of finding an axial length and a distance.

26. A retrieval apparatus that retrieves an item identical with or similar to a designated one from a memory unit storing a plurality of objects, comprising: a multidimensional index generation unit for dividing a multidimensional space into a plurality of areas to generate a multidimensional index in association with the divided areas in order to specify a predetermined object in said multidimensional space, said multidimensional index generation unit comprising reference regular simplex arranging portion adapted to arrange a regular simplex to be a reference in a certain position in said multidimensional space and sphere arranging portion adapted to arrange a sphere at a vertex of said regular simplex arranged by said reference regular simplex arranging portion and dividing said multidimensional space by the sphere; and a retrieval unit for using a multidimensional index generated by said multidimensional index generation unit to retrieve said object.

27. The retrieval apparatus according to claim 26, wherein said multidimensional index generation unit comprises an approximate information preparation unit for, in retrieving a predetermined point in a multidimensional space that is registered as a position in said multidimensional space, preparing approximate information that is obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to positional information concerning said registered point in said multidimensional space.

28. The retrieval apparatus according to claim 26, wherein said approximate information preparation unit comprises: vector setting portion adapted to set a set of direction vectors representing a direction in said multidimensional space and, at the same time, setting a predetermined direction vector corresponding to said predetermined point using at least a part of said set of direction vectors; axial length calculating portion adapted to find a length from an origin of said set predetermined direction vector to a closest point from the point on said predetermined direction vector as an axial length; distance calculating portion adapted to find a length from the point to the closest point on said direction vector as a distance; and approximate information forming portion adapted to form said approximate information based on a predetermined direction vector set by said vector setting portion, an axial length calculated

by said axial length calculating portion and a distance calculated by said distance calculating portion.

29. A multidimensional index generation program for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas, which is stored in a computer readable storage medium, said multidimensional index generation program causing a computer to execute: a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in said multidimensional space; and a sphere arrangement step of arranging a sphere at a vertex of said regular simplex arranged by said reference regular simplex arrangement step and dividing said multidimensional space by said sphere.

30. An approximate information preparation program for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to said positional information concerning said registered point in said multidimensional space, which is stored in a computer readable storage medium, said approximate information preparation program causing a computer to execute: a vector setting step of setting a set of direction vectors representing a direction in said multidimensional space and, at the same time, setting a predetermined direction vector corresponding to said predetermined point using at least a part of said set of direction vectors; a step of finding a length from an origin of said set predetermined direction vector to a closest point from the point on said predetermined direction vector as an axial length and finding a length from the point to the closest point on said direction vector as a distance; and an approximate information forming step of forming the approximate information based on a predetermined direction vector set by said vector setting step, a calculated axial length and a calculated distance calculated by said step of finding an axial length and a distance.

Previous Doc Next Doc Go to Doc#

L17: Entry 5 of 18 File: PGPB Feb 5, 2004

DOCUMENT-IDENTIFIER: US 20040024738 A1
TITLE: Multidimensional index generation apparatus, multidimensional index generation method, approximate information preparation apparatus, approximate information preparation method, and retrieval apparatus

Abstract Paragraph:
To cluster a space efficiently even in a high dimension, and realize high speed in a high dimension, and to perform similarity retrieval that can store approximate information without any waste and in a short form, can reduce an overall storage space, and can reduce the number of times of access of processing such as retrieval. There is provided a multidimensional index generation apparatus for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided area, which arranges a regular simplex to be a reference in a certain position in the multidimensional space, arranges spheres at vertexes of the arranged regular simplex, and divides the multidimensional space by the spheres.

Summary of Invention Paragraph:
[0002] The present invention relates to a retrieval apparatus that is arranged to retrieve an item similar to or identical with a designated one, and a multidimensional index generation apparatus, a multidimensional index generation method, an approximate information preparation apparatus, and an approximate information preparation method that are applied to the retrieval apparatus. In particular, the present invention relates to those apparatuses and methods that are arranged to be able to perform the retrieval and similarity retrieval at a high speed.

Summary of Invention Paragraph:
[0006] When a retrieval apparatus for performing similarity retrieval is constituted by using a computer, a plurality of characteristics (e.g., color and shape) of an object like an image (hereinafter referred to as object) are generally extracted as numerical values and are represented as points in a multidimensional space having a set of the numerical values as coordinates. If n characteristics are extracted, the characteristics are represented as points in an n-dimensional space. A dimension ranges from a few dimensions to as large as several hundred dimensions. A point corresponding to an object is correctly referred to as an object point. However, if it is not likely that misunderstanding occurs, it is simply referred to as a point.

Summary of Invention Paragraph:
[0007] A point in a multidimensional space is also considered to be a position vector from an origin. A vector is a concept of an arrow from a start point to an end point and is a concept having both a direction and a length. A start point of a vector does not have to be a specific point. However, a specific point such as an origin is considered to be a start point and a vector representing a position of the point is specifically referred to as a position vector. When one wishes to grasp a point specifically as a position vector, that is, when one wishes to grasp it as a volume having a direction and a length, a term of vector is used. In a case of an object point, a vector is referred to as an object vector or simply as a vector.

Summary of Invention Paragraph:
[0011] A simplest method of the similarity retrieval is a sequential method of checking for all points in a multidimensional space whether the points are close to a designated point. However, this method takes an extremely long time because all point records are accessed. Thus, a large number of methods are proposed which prepare an index called a multidimensional index other than the point records and use this index to reduce accesses to point records.

Summary of Invention Paragraph:

[0012] In the multidimensional index, a space is generally divided into a plurality of areas by a solid such as a cuboid or a sphere. An area occupied by this solid is referred to as a cluster. Then, points included in the cluster are managed collectively. In the SS-tree method `see [White96] D. A. White et al.: "Similarity Indexing with the SS-tree", Proc. 12.sup.th ICDE, pp.516-523 (1996)`, a sphere is a cluster and a space is divided into a plurality of spheres. In the R*tree method `see [Beckmann90] N. Beckmann: "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. SIGMOD 1990, pp.322-331 (1990)`, a cluster is a cuboid and a space is divided into a plurality of cuboids. At the time of retrieval, only clusters close to a designated point are retrieved, whereby the number of times of access to point records is reduced. In many cases, information in a cluster is accessed collectively during processing. Therefore, the information is desirably stored on a secondary memory collectively. Bringing information into this state is referred to as clustering.

Summary of Invention Paragraph:
[0013] Information on a cluster or information on points included in the cluster is managed within a multidimensional index as a retrieval record (index record). Although the index record may be referred to as an index record, it is referred to as a index record for simplicity. Any multidimensional index has this index record inside it.

Summary of Invention Paragraph:
[0020] Incidentally, various methods have been proposed conventionally concerning the multidimensional index `see [Gaede98] V. Gaede et al.: "Multidimensional Access Methods", ACM Computing Surveys, Vol. 30, No. 2, (June 1998)`. These methods are roughly classified as follows:

Summary of Invention Paragraph:
[0028] An index has a hierarchical structure. By dividing a multidimensional space into partial areas hierarchically, a retrieval range is limited to realize speed-up.

Summary of Invention Paragraph:
[0046] Incidentally, in the multidimensional index, it is important to reduce access to a point record or an index record. As a method for this, a method of reducing the number of times of access is proposed which extracts shorter information from the point record or the index record (this information is referred to as approximate information), and uses the approximate information to determine whether it is necessary to access the point record or the index record. Reducing the number of times of access using this method is referred to as filtering. Finding an approximate position is referred to as approximation. If this is compared to a map, it corresponds to finding information indicating an approximate position such as a country, a prefecture or a city as opposed to an address as accurate as a number of street.

Summary of Invention Paragraph:
[0053] Here, it is assumed that an approximate object point according to rectangular coordinates exists in a cuboid. When this cuboid is divided at an equal interval for each coordinate axis, the cuboid can be divided into a plurality of partial cuboids. This partial cuboid is referred to as a cell. Then, information on which cell an object point belongs to is assumed to be approximate information. Compared with representing an object point with accurate coordinates, since it is not seen where in the cell the object point exists, the information is approximate. However, it can be represented with far less volume for that as information.

Summary of Invention Paragraph:
[0056] (Multidimensional Index)

Summary of Invention Paragraph:
[0057] As the Internet and input apparatuses (scanner, digital camera) widespread, both the number and a volume of multimedia data are sharply increasing. As the number of multimedia data increases, a technique for retrieving the data is naturally required. In particular, in the case of multimedia, there are high expectations for similarity retrieval based on its contents. In addition, since the number of retrieval objects increases, a high-speed retrieval is required. In research and development of a multidimensional index, importance is often attached to this speed-up. A performance of the similarity retrieval is significantly affected by the number of times of input/output, and it is a key point how to reduce this number of times of input/output.

Summary of Invention Paragraph:
[0058] If the number of times of input/output is reduced, two points concerning a space efficiency and adaptability in a high dimension are important. As to the space efficiency, it is important to make a cluster and approximate information on the multidimensional index as compact as possible and reduce the number of times of input/output. As to the adaptability in a high dimension, accuracy of the similarity retrieval can be generally attained by increasing the number of characteristic volumes, that is, making a dimension of the multidimensional space higher. However, when a dimension is increased to as high as several tens dimensions to several hundred dimensions, as introduced in [Katayamaol] `see Katayama Norio et al.: "Index Technique for Similarity Retrieval", Joho shori (Information Processing) Vol. 42, No. 10, pp. 958-964, (October 2001)`, a phenomenon called a curse of dimensionality occurs, and a performance of similarity retrieval generally falls. According to the curse of dimensionality, it is known that problems such as the similarity retrieval and multivariate analysis become difficult in a high dimension. These problems are collectively referred to as the curse of dimensionality. As a concrete example, when points are uniformly distributed in a multidimensional space, a phenomenon that, in view of a certain point, other points gather near a spherical surface with the point as a center. That is, there is little difference of distances.

Summary of Invention Paragraph:
[0061] In the conventional technique, the inside of a cuboid is approximated by rectangular coordinates. On the other hand, there are provided a large number of multidimensional indexes using a sphere (see [Katayama97] `N. Katayama el al.: "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries", Proc. SIGMOD 1997, pp.369-380 (1997)`, [White96]). If it is attempted to approximate points in a sphere by a method according to a cuboid, the inside of a cube circumscribing the sphere is represented by rectangular coordinates as shown in FIG. 37. For simplicity, the case of a two-dimension will be described first. If it is attempted to approximate a point with the conventional method in the two-dimension, a result shown in FIG. 38 is obtained. Here, a point is approximated with total 256 square cells that are divided equally vertically and horizontally into sixteen pieces. A cell including a point P can be represented as (5,3). As the vertical and horizontal parts are divided into sixteen pieces, each can be represented as a bit, and total can be represented as 8 bit. However, in this case, areas such as (1,1) and (2,0) are outside the sphere. There are 40 or more such partial squares in total. That is, waste occurs in representation. This waste is less in the case of two-dimension. Next, a case of a high dimension will be described.

Summary of Invention Paragraph:
[0065] (Multidimensional Index and Filtering)

Summary of Invention Paragraph:
[0066] A database system, in particular, a relational database is becoming complicated according to expansion of a specification of SQL. As stated in [Chaudhuri00] `see S. Chaudhuri et al.: "Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System", Proc. of Intl. Conf. of Very Large Database Systems, (2000)`, since functions of a database system are expanded and optimization is complicated, maintenance, management, performance estimate and the like are becoming difficult and maintenance costs and management costs are increasing. Thus, simplification is desired. In a page based method of controlling a page that is a unit of input/output by oneself, although clustering is easily controlled, a kernel part of the database system should be manipulated. The database system is becoming huge and complicated, and a lot of studies of an expansion database for facilitating such expansion of functions are performed. However, in an actual development side, if such expansion is performed, a large amount of costs are incurred including those for tests and maintenance as an actual situation. This seems to be a reason why a method of multidimensional index is not put into practice in spite of the fact that many methods of multidimensional index are proposed.

Summary of Invention Paragraph:
[0068] Similarly, if a method for a multidimensional index can be realized on a database system, it becomes easy to put the method into practice. If it is prepared based on the standard such as SQL, it also becomes possible to run it on many existing database systems. In this case, since no manipulation can be applied to a page, the application is realized by record manipulation, that is, the application is based on a record. Although in the record based application the application is easy to realize, since clustering cannot be controlled generally, it is required to reduce the number of times of access to a record.

Summary of Invention Paragraph:
[0069] The present invention has been achieved in order to solve the above-mentioned problems, and it is an object of the present invention to provide a multidimensional index generation apparatus, a multidimensional index generation method, an approximate information preparation apparatus, an approximate information preparation method and a retrieval apparatus that can divide a sphere efficiently, can realize efficient use of a storage space, can attain speed-up of retrieval processing, and can establish the inside of a sphere with shorter approximate information to realize efficiency of a storage space and cost reduction, thereby being able to easily perform establishment of a system.

Summary of Invention Paragraph:
[0070] In order to solve the above-mentioned problem, the present invention provides a multidimensional index generation apparatus for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas in order to specify a predetermined point in the multidimensional space, which includes reference regular simplex arrangement means for arranging a regular simplex to be a reference in a certain position in the multidimensional space, and sphere arrangement means for arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arrangement means and dividing the multidimensional space by the sphere. In an embodiment of the present invention, the reference regular simplex arrangement means and the sphere arrangement means are constituted by cooperation of a control device 11, a sphere generation device 12 and a point generation device 13.

Summary of Invention Paragraph:
[0071] In addition, the multidimensional index generation apparatus of the present invention further includes connection regular simplex arrangement means for arranging a plurality of regular simplexes by connecting the regular simplex to another regular simplex with the same size as the regular simplex once or more such that surfaces of both the regular simplexes join each other, and the sphere arrangement means is characterized by dividing the multidimensional space by arranging a sphere on a vertex of the regular simplex arranged by the reference regular simplex arrangement means as well as vertexes of the plurality of regular simplexes arranged by the connection regular simplex arrangement means.

Summary of Invention Paragraph:
[0072] Further, in the multidimensional index generation apparatus of the present invention, the reference regular simplex arrangement means or the connection regular simplex arrangement means is characterized by arranging a further regular simplex for a sphere arranged by the sphere arrangement means and dividing the sphere in a hierarchical manner by the sphere arrangement means arranging a further sphere at a vertex of the further regular simplex.

Summary of Invention Paragraph:
[0073] In the multidimensional index generation apparatus of the present invention, the multidimensional space is a sphere as a partial space, and the reference regular simplex arrangement means may also be characterized by arranging the regular simplex to be a reference such that the center of gravity of the regular simplex to be a reference coincides with a center of the sphere.

Summary of Invention Paragraph:
[0074] In addition, in the multidimensional index generation apparatus, the multidimensional space is a sphere as a partial space, and the reference regular simplex arrangement means may also be characterized by arranging the regular simplex to be a reference such that the center of gravity of the regular simplex to be a reference coincides with a center of a substantial sphere by a point included in the sphere of the multidimensional space.

Summary of Invention Paragraph:
[0075] Moreover, the multidimensional index generation apparatus may also be characterized by including judging means for judging the number of vectors included in a sphere and vector holding means for, based on a result of judgment by the judging means, if the number of vectors included in the sphere is small, holding the vectors as they are without turning the vectors into a sphere. Note that this vector holding means is also constituted by cooperation of the control device 11, the sphere generation device 12 and the point generation device 13.

Summary of Invention Paragraph:

[0076] Moreover, the multidimensional index generation apparatus may also be characterized by including clustering means for performing clustering by arranging identifiers specifying the object point in hierarchy based on the divided sphere.

Summary of Invention Paragraph:
[0077] In addition, the present invention provides a multidimensional index generation method of dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided area, which includes a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in the multidimensional space and a sphere arrangement step of arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arrangement step and dividing the multidimensional space by the sphere.

Summary of Invention Paragraph:
[0079] In addition, the present invention provides an approximate information preparation apparatus for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to the positional information concerning the registered point in the multidimensional space, which includes vector setting means for setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, axial length calculating means for finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length, distance calculating means for finding a length from the point to the closest point on the direction vector as a distance, and approximate information forming means for forming the approximate information based on a predetermined direction vector set by the vector setting means, an axial length calculated by the axial length calculating means and a distance calculated by the distance calculating means. Further, the approximate information preparation apparatus corresponds to the approximate information generation device in the embodiment of the present invention, and the axial length calculating means, the distance calculating means and the approximate information forming means are constituted by cooperation of an arithmetic unit such as a CPU and software.

Summary of Invention Paragraph:
[0085] Moreover, in the approximate information preparation apparatus of the present invention, the vector setting means may be characterized by setting the direction vector based on each coordinate value in the case in which a predetermined point in the multidimensional space is represented by rectangular coordinates and, at the same time, setting the predetermined direction vector.

Summary of Invention Paragraph:
[0086] In addition, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by arranging a regular simplex in the multidimensional space, and using vertex vectors as a vector from the center of gravity of the regular simplex to a vertex of all or at least a part of the regular simplex to set the direction vector and, at the same time, setting the predetermined vector.

Summary of Invention Paragraph:
[0089] Moreover, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by including means for arranging a regular simplex in the multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from the center of gravity of the regular simplex to the vertex of the regular simplex, and finding vectors g(1), g(2), . . . , g(k) as

Summary of Invention Paragraph:
[0091] Moreover, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by including means for arranging a regular simplex in the multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from the center of gravity of the regular simplex to the vertex of the regular simplex,

and finding vectors g(1), g(2), , g(k) as

Summary of Invention Paragraph:
[0103] In addition, in the approximate information preparation apparatus of the present invention, the vector setting means is characterized by setting a direction vector by recursively dividing a dimension of a vector obtained by normalizing an object vector as a vector representing the predetermined point, constituting an identifier using a ratio of length, and assigning bits such that a surface area of a divided sphere and the number according to a bit assigned to a divided vector are proportional to each other.

Summary of Invention Paragraph:
[0104] In addition, the present invention provides an approximate information preparation method of, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to the positional information concerning the registered point in the multidimensional space, which includes a vector setting step of setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, a step of finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length and finding a length from the point to the closest point on the direction vector as a distance, and an approximate information forming step of forming the approximate information based on a predetermined direction vector set by the vector setting step, a calculated axial length and a calculated distance calculated by the step of finding an axial length and a distance.

Summary of Invention Paragraph:
[0106] In addition, the present invention provides a retrieval apparatus that retrieves an item identical with or similar to a designated one from a memory unit storing a plurality of objects, which includes a multidimensional index generation unit for dividing a multidimensional space into a plurality of areas to generate a multidimensional index in association with the divided areas in order to specify a predetermined object in the multidimensional space, the multidimensional index generation unit including reference regular simplex arranging means for arranging a regular simplex to be a reference in a certain position in the multidimensional space and sphere arranging means for arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arranging means and dividing the multidimensional space by the sphere, and a retrieval unit for using a multidimensional index generated by the multidimensional index generation unit to retrieve the object.

Summary of Invention Paragraph:
[0107] In addition, in the retrieval apparatus of the present invention, the multidimensional index generation unit is characterized by including an approximate information preparation unit for, in retrieving a predetermined point in a multidimensional space that is registered as a position in the multidimensional space, preparing approximate information that is obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to positional information concerning the registered point in the multidimensional space.

Summary of Invention Paragraph:
[0108] Moreover, in the retrieval apparatus of the present invention, the approximate information preparation unit may be characterized by including vector setting means for setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, axial length calculating means for finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length, distance calculating means for finding a length from the point to the closest point on the direction vector as a distance, and approximate information forming means for forming the approximate information based on a predetermined direction vector set by the vector setting means, an axial length calculated by the axial length calculating means and a distance calculated by the distance calculating means.

Summary of Invention Paragraph:
[0110] Further, in this embodiment, there is disclosed a multidimensional index generation program for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas, which is stored in a computer readable storage medium, the multidimensional index generation program causing a computer to execute a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in the multidimensional space and a sphere arrangement step of arranging a sphere at a vertex of the regular simplex arranged by the reference regular simplex arrangement step and dividing the multidimensional space by the sphere.

Summary of Invention Paragraph:
[0111] In addition, in this embodiment, there is disclosed an approximate information preparation program for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning the registered point in the multidimensional space in order to reduce the number of times of access to the positional information concerning the registered point in the multidimensional space, which is stored in a computer readable storage medium, the approximate information preparation program causing a computer to execute a vector setting step of setting a set of direction vectors representing a direction in the multidimensional space and, at the same time, setting a predetermined direction vector corresponding to the predetermined point using at least a part of the set of direction vectors, a step of finding a length from an origin of the set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length and finding a length from the point to the closest point on the direction vector as a distance, and an approximate information forming step of forming the approximate information based on a predetermined direction vector set by the vector setting step, a calculated axial length and a calculated distance calculated by the step of finding an axial length and a distance. In this case, the computer readable medium includes portable storage media such as a CD-ROM, a flexible disk, a DVD disk, a magneto-optical disk and an IC card, a database for holding a computer program, or other computers and a database therefor, and a transmission medium on a line.

Brief Description of Drawings Paragraph:
[0137] FIG. 19 is a flow chart showing a flow at the time of generation of a multidimensional index;

Detail Description Paragraph:
[0162] The similarity retrieval apparatus of the embodiment consists of a generation apparatus 1 for performing generation and update of a multidimensional index, a retrieval apparatus (similarity retrieval apparatus) 2 for using a generated multidimensional index to perform similarity retrieval and, at the same time, using approximate information to perform filtering processing, and a database 3.

Detail Description Paragraph:
[0165] The database 3 is constituted by a sphere relation database 31 for storing a sphere relation and a point relation database 32 for storing a point relation. Preparation (establishment) of a multidimensional index, preparation of approximate information and similarity retrieval (retrieval), all of which are executed in this apparatus, will be hereinafter described.

Detail Description Paragraph:
[0166] I. Establishment of a Multidimensional Index

Detail Description Paragraph:
[0167] In a multidimensional index method, first, how to constitute a multidimensional index is important. It will be hereinafter described how a multidimensional index is established by a generation apparatus.

Detail Description Paragraph:
[0184] Here, a vector from the center of gravity of the regular simplex in question to a point is referred to as a center of gravity to point vector. In addition, a surface vector with the smallest angle with respect to the center of gravity to point vector among n+1 surface vectors of the regular simplex is found, and a new regular simplex is connected to a surface

corresponding to this surface vector such that surfaces match each other well. In this way, new regular simplexes are created one after another. This is called growth of a regular simplex. Every time a regular simplex grows, a newly generated regular simplex approaches the point. In the basic division, the number of spheres is n+1 at the maximum, and a size of a sphere is generally large. A radius of a sphere is restricted by a distribution of a set of points.

Detail Description Paragraph:
[0185] Next, a method of dividing a space by a sphere of an arbitrary radius will be described more generally. FIG. 4 is a diagram showing the case of the two-dimension. In this figure, as it is well known, a two-dimensional space (plane) is covered by circles of the same radius without a gap and with least overlapping. FIG. 5 is a diagram in which a part of FIG. 4 is extracted and centers of the circles are connected by lines. When observed well, in this figure, regular triangles are regularly arranged and circles are arranged at vertexes thereof. This arrangement of regular triangles is attained by placing one regular triangle to be a reference first and subsequently attaching regular triangles such that sides of the regular triangles match each other.

Detail Description Paragraph:
[0186] In the case of the three-dimension, this cannot be attained so simply as in the case of the two-dimension. This is because, if a regular tetrahedron of the same size is attached to one regular tetrahedron to be a reference as in the case of the two-dimension such that surfaces match each other, a gap is generated between the regular tetrahedrons. It is known that, if five regular tetrahedrons are connected so as to draw a circle, a gap of approximately 10 degrees is generated between a first regular tetrahedron and a last regular tetrahedron. If it is attempted to draw more circles, the circles do not completely match the regular tetrahedron to be a reference but intersect it. That is, the three-dimensional space cannot be covered by regular tetrahedrons without overlapping as in the case of the two-dimension. In the three-dimension, the question of what is an arrangement without a gap and with least overlapping has been unsolved for nearly 400 years. Recently, it seems that a most densely filled structure (usually, a method adopted in filling balls in a box) has been proved to be most suitable.

Detail Description Paragraph:
[0195] <5> A surface vector with the smallest angle with respect to the center of gravity to point vector among n+1 surface vectors of the regular simplex is found, and a new regular simplex is connected to a surface corresponding to this surface vector such that surfaces match each other well.

Detail Description Paragraph:
[0234] <1> The center of gravity of a reference regular simplex a is matched with the center of the sphere Sd. A method for obtaining a radius of the reference regular simplex .sigma. is described below. A set of points included in the sphere Sd is assumed to be as follows:

Detail Description Paragraph:
[0258] <1> The center of gravity of a reference regular simplex .sigma. is matched with the center of the sphere Sd. A radius of the reference regular simplex .sigma. is made equal to the radius of the sphere Sd or larger than a radius of a root sphere. A set of points included in the sphere Sd is assumed to be as follows:

Detail Description Paragraph:
[0267] 3.3) A Method of Matching the Center of gravity of Data with the Center of Gravity of a Reference Regular Simplex

Detail Description Paragraph:
[0268] The present invention has been described on the premise that the center of gravity of the reference regular simplex is matched with a center of a parent sphere for ease of description. However, points included in the parent sphere are not always distributed around the center of the parent sphere. The points are likely to gather in a specific part of the parent sphere. In this case, the parent sphere is likely to be divided into a small number of spheres (one child sphere in the worst case). Thus, a method is possible which sets the center of gravity of a set of points included in the parent sphere, that is, a center of a substantial sphere as a center of the reference regular simplex. In this case, with the above-mentioned method of the basic division, a situation may occur in which a point is not included (left out)

in any child sphere. This is because a parent sphere is divided without any gap by the basic division only when a center of the sphere and the center of gravity of a reference regular simplex coincide with each other. Therefore, in this method, the extended division is always used. Note that it is possible to use the basic division unnaturally by enlarging a radius of a reference regular simplex. However, this is unadvisable because a radius of a child sphere is larger than a parent sphere in this case.

Detail Description Paragraph:
[0296] The storage structure of a relation has been described. Here, it will be described how an index is generated as a whole using the storage structure. FIG. 19 shows an entire flow diagram of operations of a multidimensional index generation apparatus.

Detail Description Paragraph:
[0301] The preparation of a multidimensional index by spheres has been described. More speed-up can be realized by further adding filtering by approximation to this method. This method of approximation will be described first.

Detail Description Paragraph:
[0304] A situation in which points are distributed in a sphere with a certain point as a center will be hereinafter considered. The center may be an arbitrary point but is assumed to coincide with an origin of a multidimensional space in order to simplify descriptions. This sphere is referred to as an object sphere in that object points are distributed in its inside. A radius of the object sphere may be arbitrary but is assumed to be 1 without losing generality in order to simplify descriptions as well. A sphere with a radius 1 is also referred to as a unit sphere.

Detail Description Paragraph:
[0344] A regular simplex will be considered. The center of gravity of this regular simplex is matched with an origin of an object sphere. A length to each vertex of the regular simplex from the center of gravity is assumed to be 1. Therefore, this regular simplex internally contacts the object sphere (because a radius of the object sphere is assumed to be 1 without losing generality). A vector from this center of gravity to each vertex is referred to as a vertex vector, and this vertex vector is assumed to be a direction vector. Therefore, first, n+1 direction vectors equivalent to the number of vertexes are created. These vectors are assumed to be as follows:

Detail Description Paragraph:
[0496] The method of approximating an object point has been described. However, in general, this method can approximate not only an object point but also a point. In particular, since a center of a sphere is also a point, it has a significant meaning to approximate a center. This is because, if a center can be approximated, the sphere itself can also be approximated by adding information on a radius to it. It is described above that the method of using a sphere as a cluster in a multidimensional index is used. With these methods, it is judged whether or not a sphere that is a cluster intersects vicinity and, if not intersecting, the number of times of access to a point vector or an index vector is reduced utilizing the advantage that the inside of the sphere may not be checked. If it is checked whether or not the sphere intersects the vicinity, an index record corresponding to the sphere is accessed to make the judgment from information on coordinates and a radius of its center. That is, a vector corresponding to the sphere must be accessed. However, when the sphere has been approximated, if it is found that the sphere does not intersect the vicinity from the approximate information without accessing an index record, it becomes unnecessary to access the index record.

Detail Description Paragraph:
[0502] Next, application of the above-mentioned approximation to the multidimensional index of the present invention will be described.

Detail Description Paragraph:
[0510] (b) Method Applied to a Multidimensional Index Using a Sphere

Detail Description Paragraph:
[0511] As described before, several multidimensional indexes using a sphere have been proposed. More specifically, a multidimensional index is used in SS-tree and SR-tree, and in A-tree partially. SS-tree is a first method using a sphere and is known as a high speed method.

Moreover, faster methods such as SR-tree and A-tree have been proposed which are improvements of SS-tree. In the multidimensional index using a sphere, an object point is divided by a plurality of spheres including the object point and it is judged whether or not the sphere intersects vicinity, whereby, if the sphere does not intersect the vicinity, reduction of the number of times of access to a point record is realized by utilizing the fact that it is unnecessary to check an object point included in the sphere. A set of direction vectors is decided with respect to an object point in the sphere considering that a center of the sphere is a center of the object sphere, whereby the method of the present invention can be applied. In addition, with respect to a sphere, approximate information on a sphere according to the present invention is also stored in an index record of a corresponding sphere and filtered at the time of retrieval, whereby it becomes possible to reduce the number of times of access to the index record corresponding to the sphere.

Detail Description Paragraph:
[0543] Since approximate information is used only for representing points in a sphere, points in the sphere can be approximated with less approximate information. Therefore, by applying the present invention to a multidimensional index or the like using a sphere, it becomes possible to realize similarity retrieval with less costs for reading approximate information.

Detail Description Paragraph:
[0548] As described above in detail, according to the present invention, there is an effect that a multidimensional index generation apparatus, a multidimensional index generation method, an approximate information preparation apparatus, an approximate information preparation method and a retrieval apparatus can be provided, which can divide a sphere efficiently, can realize efficiency of a storage space, can attain high speed of retrieval processing, and can establish the inside of a sphere with short approximate information to realize efficiency of a storage space and realize cost reduction, thereby performing similarity retrieval at a high speed and, at the same time, establishing apparatuses at low costs and easily.

CLAIMS:

1. A multidimensional index generation apparatus for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas in order to specify a predetermined point in the multidimensional space, comprising: reference regular simplex arrangement portion adapted to arrange a regular simplex to be a reference in a certain position in said multidimensional space; and sphere arrangement portion adapted to arrange a sphere at a vertex of said regular simplex arranged by said reference regular simplex arrangement portion and dividing said multidimensional space by said sphere.

2. The multidimensional index generation apparatus according to claim 1, further comprising connection regular simplex arrangement portion adapted to arrange a plurality of regular simplexes by connecting said regular simplex to another regular simplex with the same size as said regular simplex once or more such that surfaces of both the regular simplexes join each other, wherein said sphere arrangement portion divides said multidimensional space by arranging a sphere on a vertex of said regular simplex arranged by said reference regular simplex arrangement portion as well as vertexes of said plurality of regular simplexes arranged by said connection regular simplex arrangement portion.

3. The multidimensional index generation apparatus according to claim 2, wherein said reference regular simplex arrangement portion or said connection regular simplex arrangement portion arranges a further regular simplex for a sphere arranged by said sphere arrangement portion and divides said sphere in a hierarchical manner by said sphere arrangement portion arranging a further sphere at a vertex of the further regular simplex.

4. The multidimensional index generation apparatus according to claim 1, wherein said multidimensional space is a sphere as a partial space, and said reference regular simplex arrangement portion arranges said regular simplex to be a reference such that the center of gravity of said regular simplex to be a reference coincides with a center of said sphere.

5. The multidimensional index generation apparatus according to claim 1, wherein said multidimensional space is a sphere as a partial space, and said reference regular simplex arrangement portion arranges said regular simplex to be a reference such that the center of gravity of said regular simplex to be a reference coincides with a center of a substantial

sphere by a point included in said sphere of said multidimensional space.

6. The multidimensional index generation apparatus according to claim 1, further comprising: judging portion adapted to judge the number of vectors included in a sphere; and vector holding portion adapted to, based on a result of judgment by said judging means, if the number of vectors included in said sphere is small, hold the vectors as they are without turning the vectors into a sphere.

7. The multidimensional index generation apparatus according to claim 1, further comprising clustering portion adapted to perform clustering by arranging identifiers specifying said object point in hierarchy based on said divided sphere.

8. A multidimensional index generation method of dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided area, comprising: a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in said multidimensional space; and a sphere arrangement step of arranging a sphere at a vertex of said regular simplex arranged by said reference regular simplex arrangement step and dividing said multidimensional space by said sphere.

9. An approximate information preparation apparatus for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to the positional information concerning said registered point in said multidimensional space, comprising: vector setting portion adapted to set a set of direction vectors representing a direction in said multidimensional space and, at the same time, to set a predetermined direction vector corresponding to said predetermined point using at least a part of the set of direction vectors; axial length calculating portion adapted to find a length from an origin of said set predetermined direction vector to a closest point from the point on the predetermined direction vector as an axial length; distance calculating portion adapted to find a length from the point to the closest point on the direction vector as a distance; and approximate information forming portion adapted to form said approximate information based on a predetermined direction vector set by said vector setting portion, an axial length calculated by said axial length calculating portion and a distance calculated by said distance calculating portion.

15. The approximate information preparation apparatus according to claim 9, wherein said vector setting portion sets said direction vector based on each coordinate value in the case in which a predetermined point in said multidimensional space is represented by rectangular coordinates and, at the same time, sets said predetermined direction vector.

16. The approximate information preparation apparatus according to claim 9, wherein said vector setting portion arranges a regular simplex in said multidimensional space, and uses vertex vectors as a vector from the center of gravity of said regular simplex to a vertex of all or at least a part of said regular simplex to set said direction vector and, at the same time, sets the predetermined vector.

19. The approximate information preparation apparatus according to claim 16, wherein said vector setting portion comprises: a portion for arranging a regular simplex in said multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from the center of gravity of said regular simplex to the vertex of the regular simplex, and finding vectors g(1), g(2), . . . , g(k) as g(1)=v(i(1)) g(2)=(v(i(1))+v(i(2))/2 . . . g(k)=(v(i(1))+v(i(2))+ . . . +v(i(k)))/k; a portion for finding a vector g=n((g(1)+g(2)+ . . . +g(k))/k) that is found by normalizing vectors to centers of gravity of g(1), g(2), . . . , g(k) to set them as direction vectors; and a portion for using numbers i(1), i(2), . . . , i(k) of vertex vectors as said predetermined vector to set said predetermined vector.

20. The approximate information preparation apparatus according to claim 16, wherein said vector setting portion comprises: a portion for arranging a regular simplex in said multidimensional space, selecting k(k<=n) vectors v(i(1)), v(i(2)), . . . , v(i(k)) in order from one having a smallest argument with an object vector out of vertex vectors as vectors from

the center of gravity of the regular simplex to the vertex of the regular simplex, and finding vectors g(1), g(2), . . . , g(k) as g(1)=n(v(i(1))) g(2)=n((v(i(1))+v(i(2))/2) . . . g(k)=n((v(i(1))+v(i(2))+ . . . +v(i(k)))/k); and a portion for, based on g(1), g(2), . . . , g(k), finding a vector g(i) having a smallest argument with an object vector among them, finding a vector m(j) from the origin O to a midpoint of g(j) (j.noteq.i) and g(i) as m(j)=(g(i)+g(j))/2, finding a vector group g(1), g(2), . . . , g(k) found by normalizing this m(j), and repeating this processing t times and, thereafter, setting a direction vector by finding the center of gravity g of g(1), g(2), . . . , g(k) and normalizing the center of gravity g, and setting said predetermined vector by a set of (j1, j2, . . . , jt).

24. The approximate information preparation apparatus according to claim 9, wherein said vector setting portion sets a direction vector by recursively dividing a dimension of a vector obtained by normalizing an object vector as a vector representing said predetermined point, constituting an identifier using a ratio of length, and assigning bits such that a surface area of a divided sphere and the number according to a bit assigned to a divided vector are proportional to each other.

25. An approximate information preparation method of, in retrieving a predetermined point in a multidimensional space registered as a position in said multidimensional space, preparing approximate information obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to said positional information concerning said registered point in said multidimensional space, comprising: a vector setting step of setting a set of direction vectors representing a direction in said multidimensional space and, at the same time, setting a predetermined direction vector corresponding to said predetermined point using at least a part of said set of direction vectors, a step of finding a length from an origin of said set predetermined direction vector to a closest point from the point on said predetermined direction vector as an axial length and finding a length from the point to the closest point on said direction vector as a distance; and an approximate information forming step of forming said approximate information based on a predetermined direction vector set by said vector setting step, a calculated axial length and a calculated distance calculated by said step of finding an axial length and a distance.

26. A retrieval apparatus that retrieves an item identical with or similar to a designated one from a memory unit storing a plurality of objects, comprising: a multidimensional index generation unit for dividing a multidimensional space into a plurality of areas to generate a multidimensional index in association with the divided areas in order to specify a predetermined object in said multidimensional space, said multidimensional index generation unit comprising reference regular simplex arranging portion adapted to arrange a regular simplex to be a reference in a certain position in said multidimensional space and sphere arranging portion adapted to arrange a sphere at a vertex of said regular simplex arranged by said reference regular simplex arranging portion and dividing said multidimensional space by the sphere; and a retrieval unit for using a multidimensional index generated by said multidimensional index generation unit to retrieve said object.

27. The retrieval apparatus according to claim 26, wherein said multidimensional index generation unit comprises an approximate information preparation unit for, in retrieving a predetermined point in a multidimensional space that is registered as a position in said multidimensional space, preparing approximate information that is obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to positional information concerning said registered point in said multidimensional space.

28. The retrieval apparatus according to claim 26, wherein said approximate information preparation unit comprises: vector setting portion adapted to set a set of direction vectors representing a direction in said multidimensional space and, at the same time, setting a predetermined direction vector corresponding to said predetermined point using at least a part of said set of direction vectors; axial length calculating portion adapted to find a length from an origin of said set predetermined direction vector to a closest point from the point on said predetermined direction vector as an axial length; distance calculating portion adapted to find a length from the point to the closest point on said direction vector as a distance; and approximate information forming portion adapted to form said approximate information based on a predetermined direction vector set by said vector setting portion, an axial length calculated

by said axial length calculating portion and a distance calculated by said distance calculating portion.

29. A multidimensional index generation program for dividing a multidimensional space into a plurality of areas and generating a multidimensional index in association with the divided areas, which is stored in a computer readable storage medium, said multidimensional index generation program causing a computer to execute: a reference regular simplex arrangement step of arranging a regular simplex to be a reference in a certain position in said multidimensional space; and a sphere arrangement step of arranging a sphere at a vertex of said regular simplex arranged by said reference regular simplex arrangement step and dividing said multidimensional space by said sphere.

30. An approximate information preparation program for, in retrieving a predetermined point in a multidimensional space registered as a position in the multidimensional space, preparing approximate information obtained by approximating positional information concerning said registered point in said multidimensional space in order to reduce the number of times of access to said positional information concerning said registered point in said multidimensional space, which is stored in a computer readable storage medium, said approximate information preparation program causing a computer to execute: a vector setting step of setting a set of direction vectors representing a direction in said multidimensional space and, at the same time, setting a predetermined direction vector corresponding to said predetermined point using at least a part of said set of direction vectors; a step of finding a length from an origin of said set predetermined direction vector to a closest point from the point on said predetermined direction vector as an axial length and finding a length from the point to the closest point on said direction vector as a distance; and an approximate information forming step of forming the approximate information based on a predetermined direction vector set by said vector setting step, a calculated axial length and a calculated distance calculated by said step of finding an axial length and a distance.

Previous Doc Next Doc Go to Doc#

MENU

L17: Entry 6 of 18 File: PGPB Jan 8, 2004

DOCUMENT-IDENTIFIER: US 20040006574 A1
TITLE: Methods of navigating a cube that is implemented as a relational object

Abstract Paragraph:
A method and mechanism for performing an operation based on multidimensional data in a relational database. A query is generated that includes a first set of instructions to select a portion of multidimensional data for inclusion in a relational cube and a second set of instructions to group the portion of multidimensional data in the relational cube using at least two or more levels of granularity of at least one dimension, thereby causing a relational cube to be generated. Based on a received request for an operation to be performed, the query is modified to include a third set of instructions that represent the requested operation; and the query is submitted to the relational database engine of the relational database management system. Thereafter, the operation is performed against the relational cube.

Summary of Invention Paragraph:
[0002] The present invention relates to relational database management systems and, more specifically, to techniques for storing multidimensional data in relational database management systems.

Summary of Invention Paragraph:
[0003] In the context of database systems, a "dimension" is a list of values that provide categories for data. A dimension acts as an index for identifying values of a variable. For example, if sales data has a separate sales figure for each month, then the data has a MONTH dimension. That is, the data is organized by month. A dimension is similar to a key in a relational database. Data that is organized by two or more dimensions is referred to as "multidimensional data".

Summary of Invention Paragraph:
[0004] Any item of data within a multidimensional measure can be uniquely and completely selected by specifying one member from each of the measure's dimensions. For example, if a sales measure is dimensioned by MONTH, PRODUCT, and MARKET, specifying "January" for the MONTH dimension, "Stereos" for the PRODUCT dimension, and "Eastern Region" for the MARKET dimension uniquely specifies a single value of the measure. Thus, dimensions offer a concise and intuitive way of organizing and selecting data for retrieval, updating, and performing calculations.

Summary of Invention Paragraph:
[0005] Multidimensional data may be stored in relational database systems ("ROLAP" systems) or in specialized, "multidimensional" database systems ("MOLAP" systems). Multidimensional database systems provide structures and access techniques specifically designed for multidimensional data, and therefore provide relatively efficient storage and access to multidimensional data. However, when stored in specialized multidimensional database systems, only applications that are specially built to interact with those multidimensional database systems are able to access and manipulate the data.

Summary of Invention Paragraph:
[0007] Relational database systems store data in the form of related tables, where each table has one or more columns and zero or more rows. The conventional mechanism for storing multidimensional data in a relational database system is to store the data in tables arranged in what is referred to as a star schema. In relational database systems, a star schema is distinguished by the presence of one or more fact tables and one or more dimension tables. Fact tables store measures, and contain foreign keys to dimension tables. Dimension tables store values for hierarchical dimensions. FIG. 1 illustrates an exemplary star schema with two

dimensions.

Summary of Invention Paragraph:
[0010] The data stored in fact table 106 only has two dimensions, and therefore fact table 106 only has two columns dedicated to storing foreign key values for those dimensions. In general, a fact table must dedicate one column for storing foreign key values for each of the dimensions associated with the multidimensional data stored in the fact table. Thus, a fact table that stores data associated with twenty dimensions would have to dedicate twenty columns to the storage of foreign key values.

Summary of Invention Paragraph:
[0011] An alternative approach to managing multidimensional data in a relational database involves storing the data in relational files but maintaining all multidimensional structure, metadata, administration, and access control using multidimensional database system techniques. Accessing relationally-stored data using multidimensional techniques poses numerous difficulties. For example, when all administration and access to the multidimensional data are controlled exclusively through the multidimensional database system engine, two database management systems must be administered. Further, database applications that access data using conventional relational commands (e.g. SQL commands) are unable to access the multidimensional data.

Summary of Invention Paragraph:
[0012] The approaches described above for storing multidimensional data in relational database systems demonstrate the tradeoffs made by prior approaches, which have either (1) sacrificed the benefits of multidimensional storage to enjoy the benefits of modern relational systems, such as conventional relational access, or (2) sacrificed the benefits of relational storage to attain the efficiency of multidimensional storage.

Detail Description Paragraph:
[0046] The OLAP Cube, Measures, Dimensions and Hierarchies are concepts in OLAP (On Line Analytical Processing), and they may be implemented in MOLAP (Multidimensional OLAP) engines. Such engines typically contain a Data Definition Language for defining these concepts, a language or an API to navigate through cubes during analysis, and access structures to improve performance for the OLAP queries

Detail Description Paragraph:
[0204] In the above example, the term "FIRST_VALUE" is a non-ANSI SQL term that refers to a function that selects the first row from a group, and the term "NUMTOYMINTERVAL" is a non-ANSI SQL term that refers to a function that converts a number to an INTERVAL YEAR TO MONTH literal, based upon a specified unit.

Detail Description Paragraph:
[0253] In multidimensional applications, a fact table generally consists of columns that uniquely identify a row, along with other information that serves as dependent measures or attributes. In one embodiment, a new SQL clause is provided ("spreadsheet clause"), which divides the identifying columns into "partitioning" and "dimension" columns. The dimension columns uniquely identify a row within a partition and serve as array indexes to measures within it. In certain embodiments, the spreadsheet clause includes a list of updates that represent array computation within a partition and which is a part of an SQL query block. For example, the spreadsheet clause structure may be as follows:

Detail Description Paragraph:
[0258] A reference to a cell should qualify all dimensions in the partition and can use either symbolic or positional referencing. Using a symbolic reference, a single dimension is qualified using a boolean condition, as for example:

Detail Description Paragraph:
[0293] Additionally, the INCREMENT operator applies to all data types for which addition & subtraction is supported, i.e., numeric and date types. For the latter, increments of the interval type may be used.

Detail Description Paragraph:
[0319] This function can be used to emulate the behavior of some multidimensional tools; if the

cell existed and has non-null data, leave it alone, otherwise create it (if didn't exit) and assign a value. For example:

Detail Description Paragraph:
[0325] This section provides further examples of how the SQL language is extended for implementing rules in the spreadsheet clause. In general, the columns in the spreadsheet clause (i.e. columns in PARTITION/DIMENSION BY) must form a key to the query. Users can assure that by placing the GROUP BY/DISTINCT in the query or by creating unique key constraints on the tables in the query. If these two compile time recognizable conditions are not satisfied, a uniqueness verification step will be performed during run time which will add overhead to the system, thus potentially affecting its performance.

Detail Description Paragraph:
[0359] (1) Q must have the spreadsheet clause and the clauses of M and Q must match exactly.

Detail Description Paragraph:
[0397] In one embodiment, to resolve the potential cell matching problem, the system requires that not only cardinality of the ranges are the same, but also that the shapes are the same as well. To achieve this goal, in certain embodiments, for each dimension d a check is made to determine if it qualifies the same number of rows in every range. For example, for the above data, d1 in Q28 qualifies one value in both ranges (i.e., value `B` in first and `C` in second range), and d2 qualifies three values in the first range (1, 5, 6.6) and three values in the second range (11, 12.6, 24). Thus, the shapes are the same. In certain embodiments, the verification process that ensures that the shapes match is performed at run time.

Detail Description Paragraph:
[0398] In other embodiments, potential cell matching problems are resolved by explicitly enumerating cells either using the IN construct or the INCREMENT operator. Each range can then be verified at compile time as the set of cell can be easily matched for multi-measure aggregates.

Detail Description Paragraph:
[0406] The above syntax using INCREMENT is compile time verifiable and simple to understand. However, it suffers from not being able to specify the existing data. For example, observe that `d2 between 11 and 20` includes all values of d2 in that interval. However, `d2 between 11 and 20 increment 1` allows for 10 values only: 11, 12, . . . , 30.

Detail Description Paragraph:
[0448] For the duplicated data values, pointers or links are inserted into the appropriate portions of the on-disk row structure to point to the appropriate entry in symbol table 510. On-disk row structure 502b includes a portion 544 corresponding to the customer column 522, which has a duplicated data value "Joe" in the original row 502a, that is configured in the on-disk row structure 502b to include a pointer 536 to the symbol table entry 530 that matches this duplicated data value. On-disk row structure 502b includes a portion 546 corresponding to the item column 524, which has a duplicated data value "Book" in the original row 502a, that is configured in the on-disk row structure 502b to include a pointer 538 to the symbol table entry 534 that matches this duplicated data value. On-disk row structure 502b includes a portion 548 corresponding to the price column 526, which has a duplicated data value "10" in the original row 502a, that is configured in the on-disk row structure 502b to include a pointer 540 to the symbol table entry 532 that matches this duplicated data value.

Detail Description Paragraph:
[0456] The advantage of combining multiple symbol entries in this manner is that there are sequences of column values for rows stored in the block that may match these combinations. Rather than having multiple links or pointer structures between an on-disk row structure and the individual data values in a symbol table, a single link or pointer structure can be used to point to a combined symbol entry that recursively links to multiple other entries.

Detail Description Paragraph:
[0608] In the first case, start reducing the number of rows in RSM that the compressor tries to load into a single block by repeatedly bisecting [known2fit, known2notfit] interval with the start condition known2fit=number of rows in uncompressed block, known2notfit=number of rows that we estimated but that did not fit. In the second case write a compressed block and proceed

to the next set of rows. In the third case, make a new estimation for the number of rows and continue buffering the RSM.

CLAIMS:

1. A method for performing an operation based on multidimensional data in a relational database, wherein the multidimensional data that is stored in the relational database includes a plurality of dimensions and at least one dimension of the plurality of dimensions includes a hierarchy having two or more levels of granularity, the method comprising the computer-implemented steps of: generating a query that defines a cube, the query including a first set of instructions to select a portion of said multidimensional data for inclusion in said cube and a second set of instructions to cause said cube to contain multidimensional data aggregated at each of said two or more levels of granularity of said at least one dimension; receiving a request for a first operation; based on the request, modifying the query to generate a modified query that includes a third set of instructions that represent the first operation; and submitting the modified query to a relational database engine to cause the first operation to be performed against the cube.

4. The method of claim 1, wherein the cube is a multidimensional cube.

5. The method of claim 1, wherein the cube is a multidimensional cube edge.

7. The method of claim 6, wherein the multidimensional data is stored in said star schema.

8. The method of claim 6, wherein the multidimensional data is stored in said snowflake schema.

Previous Doc Next Doc Go to Doc#

L17: Entry 7 of 18 File: PGPB Dec 25, 2003

DOCUMENT-IDENTIFIER: US 20030236789 A1
TITLE: Cost conversant classification of objects

Detail Description Paragraph:
[0020] Multidimensional attribute space: A multidimensional space is a space where each dimension represents an attribute. If a multidimensional attribute space has `D` dimensions, it is referred to as a D-dimensional space and is defined by a D-tuple of ranges:

Detail Description Paragraph:
[0039] Marginal Weight Distribution: Given a set of rules, {R.sub.1,R.sub.2, . . . ,R.sub.n}, in D-dimensional space, and P.sub.r, a partition in dimension `k`, a marginal weight distribution (MWD) can be computed for P.sub.r. The marginal weight distribution assigns relative cost weights to each interval in P.sub.r.

Detail Description Paragraph:
[0040] Since we use the concept of marginal weight distribution frequently in this patent, we will now give examples of procedures that can be employed to computed such weight distribution. For example, given a set of rules whose cost weights are known, the relative cost weight of an interval in P.sub.r is equal to the sum of the cost weights of the rules that intersect with that interval in dimension `k`.

Detail Description Paragraph:
[0041] In another example, given a set of rules whose cost weights are known, the relative cost weight of an interval in P.sub.r is equal to the sum of the fractional cost weights of the rules that intersect with that interval in dimension `k`. The fractional cost weight of a rule is defined as the cost weight of rule divided by the number of intervals in P.sub.r that the rule intersects in dimension `k`. As a further example of computing marginal weight distribution consider a set of rules with initially unknown cost weights, the cost weight of an interval in P.sub.r is equal to the relative frequency at which an object's attribute in dimension `k` falls within that interval. Thus this procedure can be used to dynamically adjust the cost weight of rules during classification.

Detail Description Paragraph:
[0042] The present invention, provides methods, apparatus and systems for building a search tree for a given set of rules. Given an object, the search tree then provide a cost conversant way of finding the rules satisfied by the object. FIG. 1 shows an embodiment of a method of constructing a search tree starting from a set of rules. The first step 110 in this process is to obtain a set of hypercubes which represent the set of rules. Step 110 includes decomposing a region in a multidimensional space into at least one hypercube. For example, a region corresponding to paying customers may be further decomposed into two hypercubes corresponding to premium and regular customers. In some embodiments, step 110 also includes combining hypercubes which result in a similar classification into a composite hypercube.

Detail Description Paragraph:
[0053] Recall that during the search tree construction, we need to partition an interior node by creating two child sub-trees. The choice of decision dimension on which to partition the node is a critical one and it involves finding a dimension with a high level of uncertainty or high level of future potential cost weight.

Detail Description Paragraph:
[0054] FIG. 4 shows an example flow chart for the process of determining the best dimension on which to partition an interior node. This process maintains two variables: (1) `d`, the best dimension found thus far (2) `U`, a performance metric. We start the process in step 410 with

`d` set to 0 and `U` set to minus infinity. A pointer `i`, which points to the current dimension under consideration, is initialized to 1. In step 420, we prepare a list T.sub.i of all possible threshold values in dimension `i`. For example, this list could be any of the following:

Detail Description Paragraph:
[0057] In Step 430, we compute a list of marginal weights W.sub.i for the dimension `i`. The marginal weight W.sub.i is computed relative to the partition of the dimension `i` created by the list of threshold values T.sub.i, where the marginal weight and partition of a dimension are defined as discussed earlier. Next, in Step 440, we compute the uncertainty U.sub.i of dimension `i` from the marginal weight distribution W.sub.1. The uncertainty is a measure of how uniformly the cost weights are distributed among different intervals of the partition under consideration. For example, computing uncertainty may include computing an entropy for dimension `i`. The entropy may be computed from a marginal weight distribution as follows:

Detail Description Paragraph:
[0060] Step 470, determines if all dimensions have been evaluated or not. If `i`<`d`, then `i` is incremented by 1 in step 480 and the process continues with Step 420. If `i`.gtoreq.`d`, (implying that all dimensions have been evaluated) processing ends with step 490 where the value of `d` is declared as the best decision dimension on which to partition node `n`.

Detail Description Paragraph:
[0076] FIG. 7 is an example flow chart of the process of finding all rules that are satisfied by an object. Given a point p=(P.sub.1, . . . , P.sub.D) in D-dimensional space, the search tree is traversed to determine all the hypercubes that contain p. If no match is found, then a default object classification is returned.

Detail Description Paragraph:
[0078] This testing generally involves the following detailed sub-steps: If the list of candidate hypercubes stored in the leaf node is empty, then search terminates without finding a match. The default object classification is returned. If it is not empty, each of the candidate hypercubes in the candidate list is considered for applicability, one by one. The applicability test for each hypercube is aided by the two lists of dimensions associated with each candidate hypercube. These are examined as follows. Let W=<[L.sub.1,U.sub.1],[L.sub.2,U.sub.2], . . . , [L.sub.D,U.sub.D],> denote a candidate hypercube. Here, L.sub.i and U.sub.i represent the lower and the upper bounds of the range in the i.sup.th dimension used to specify the hypercube. The given point p belongs to hypercube W if both of the following conditions hold:

Detail Description Paragraph:
[0081] If none of the hypercubes in the candidate list meets the above conditions then search terminates without finding a match then the default object classification is returned.

CLAIMS:

2. A method as recited in claim 1, wherein the step of obtaining a set of hypercubes includes decomposing a region in a multidimensional space into at least one hypercube.

Previous Doc Next Doc Go to Doc#

L10: Entry 5 of 8 File: USPT Dec 3, 2002

DOCUMENT-IDENTIFIER: US 6490532 B1
TITLE: Method to construct protein structures

Brief Summary Text (6):
Various methods have been described for solving the folding problem. The methods include direct and template-based methods. Direct methods try to determine the native conformation as a lowest energy point in some defined hyperspace of conformational possibilities. Template-based methods compare a sequence of unknown three-dimensional structure against a library of known three-dimensional structure and score good matches as likely folds. There is a substantive body of research literature on these methods, but successes are rare and often not reproducible. There has been a call for new computational methods that broadly explore comformational space and that are true to the details of protein structure (Dill, K. A. et al Nature Structural Biology 4:10, 1997; Karpus, M. Fold Des. 2:S69, 1997).

Detailed Description Text (7):
FIG. 5 shows the node structure of the binary-d tree. Note that the left and right child pointer pairs {L1,R1}, {L2,R2} and {L3,R3} are simply binary trees that descend down 3, 2, and 1 dimensions.

Detailed Description Text (9):
FIG. 7 shows a comparison of a sphere divided equally in both angular dimensions (a) with one divided equally in .theta. and cos.phi., (b) to produce patches of equal area. The latitudinal lines in a) appear to stay almost equally spaced throughout, while in b) they are very closely spaced near the equator, and very far apart near the poles. This compensates for the convergence of the longitudinal lines as they approach the poles, and provides a more unbiased sampling of the spherical surface than a) which tends to cluster samples near the poles, where there are simply more patches per unit surface area.

Detailed Description Text (20):
FIG. 18 shows an average trajectory graph or distribution for all residues in all proteins in the non-redundant set used (FIG. 18A) and Ala only (FIG. 18B), plotted in .theta.-cos.phi. space. Each grid square corresponds to a patch of equal area on the surface of the sphere (see FIG. 19). The .alpha.-helix peak has been truncated for clarity, and note that the graph "wraps around" in the .theta. dimension. Each peak corresponds to a different type of secondary structure, as labeled, just like in Ramachandran space. Note residues in random coil or turn conformation can actually fall anywhere on the trajectory graph, including on top of the .alpha.-helix or .beta.-sheet peaks.

Detailed Description Text (45):
The data structure used here is a binary tree in one-dimension. The difference between this structure and the oct-tree or the k-d tree (Omohundro, 1987) in higher dimensions is that it is comprised of points or vectors, rather than partitioned volumes or hyper-rectangles. It is also memory efficient requiring only 2 pointers for each dimension (2d), rather than 2.sup.2 pointers for quad-tree or oct-tree systems. A program which employs the binary-3 tree to generate polyalanine structures using a kinetic, self-avoiding random growth model is used as an example of the speed of this data structure.

Detailed Description Text (46):
The binary-d tree is illustrated in FIGS. 2A and B. The recursive tree building algorithm is shown in Listing 1 for the three-dimensional case, but is easily expanded to other dimensions.

Detailed Description Text (47):
The node in this example has two arrays. The first array Dim[ ] holds the X, Y, and Z

coordinates. It is indexed with 0 for X, 1 for Y, and 2 for Z. The array of pointers to further NODEs, Way[ ], represents the six left and right child pointers. These are indexed from 0 to 5. With these arrays and indices, the left child pointer of the nth-dimension is WAY[(n-1)*2] and the right child pointer is Way[(n-1)*2+1]. The procedure PUTIN is called with three arguments: NewNode--a pointer to the NODE to be inserted into the tree; TreeNode--a pointer to the NODE in the tree with which NewNode will be compared; and dimension--the index of the dimension at which to begin the comparison. Lines 4-14 act to select the branch from TreeNode where NewNode would be attached to the tree. The values of the coordinates are compared, starting from the dimension which was passed to the procedure. If the NewNode value is greater than the TreeNode value, the right child branch index is generated for the variable branch. If it is less, the left child index is formed. Importantly, if they are equal, then the next dimension is indexed and the comparison is repeated. (In practice, equality is evaluated using some value for the desired threshold of precision.) If all three coordinates are equal, PUTIN will not insert the node (line 15). Once a branch is selected, its pointer is tested for occupancy (line 17). If it is free, NewNode is attached to the tree (line 18). If it is occupied, TreeNode is moved to the node at that branch and PUTIN is called recursively. Once the tree of data points is formed, it can be used to test for near neighbors with a three-dimensional branch and bound process. One recursive form of the code which can execute a three-dimensional branch and bound search using the binary-d tree is listed.

Detailed Description Text (48):
Three related recursive routines are used: XNeighbor, YNeighbor, and ZNeighbor. The boundary nodes MinNode and MaxNode describe a three-dimensional rectangular probe volume, as in FIG. 3 (d). In procedure XNeighbor, lines 2 and 3 traverse the tree recursively to the leftmost node within the boundary, in the first dimension. Then the next dimension's branch and bound routine, YNeighbor, is called. Finally, lines 5 and 6 perform the rightward branching. Procedures XNeighbor and YNeighbor are identical, but have different dimension indices. Procedure ZNeighbor is also like XNeighbor, but contains the complete test for the probe volume (ZNeighbor lines 4-9) instead of line 4. This is necessary since all three coordinates must be tested simultaneously to determine if the point is contained in the rectangular prism range MinNode to MaxNode. Line 9 of ZNeighbor contains a call to a routine ReportNeighbor, which can report or list the nodes encountered in the probe volume. If ReportNeighbor builds a list of these nodes as they are encountered, they will be sorted in back to front order by the branch and bound process itself. Any test of nearest neighbors that requires a non-rectangular test volume should use the branch and bound routine with a rectangular prism shaped volume enclosing that volume. Nearest neighbors found in the volume, can be further tested with euclidean distance calculations.

Detailed Description Text (51):
Upon inspection of the branching of the binary-d tree, the discrimination is based almost entirely on the first dimension. Hence a binary tree of the first dimension's coordinates presents a similar speedup. This is the general case with well separated points with high precision coordinate values. Higher dimensions in the data structure are used only when degeneracy is detected in an earlier dimension. This depends upon the resolution used to measure equality of position. For the atomic models presented here, the resolution should be small compared to the diameter of the atoms. The smaller the resolution, the less the higher dimensions of the tree are used. A binary-d tree with a coarse resolution (e.g. half an atomic diameter) can miss some near neighbors.

Detailed Description Text (52):
The binary-d tree's higher dimensions could be used much more efficiently if the resolution was imposed by a grid or lattice. In the case of integer three-dimensional space, the process of insertion of a point into a binary-3 tree reveals any prior occupant of that point without the branch and bound process. A binary-2 tree should provide sufficient segregation for nearest neighbors searches with most three-dimensional atomic data. A binary-4 tree could be used to encode time as well as three-dimensional position, allowing for time and space resolution of nearest neighbors, such as those important in celestial mechanics.

Detailed Description Text (53):
Despite the inefficiency in the use of the higher dimension pointers of this tree structure, a node in a binary-d tree has 2 fewer pointers than the corresponding oct-tree node. In addition, this tree is completely independent of previous partitions, unlike the oct-tree. A node can be removed, and its child nodes re-linked to its parent node, as with a binary tree. Hence the

tree can change dynamically, which is useful for localized or segmental motions. Optimizations such as tree balancing can also benefit this data structure (Knuth, 1973). The branch-and-bound algorithm can rapidly return a list of points in a given volume in back-to-front order. This means that hidden-line removal, clipping, and ray tracing could be implemented with the binary-d tree.

Detailed Description Text (64):
During the random walk and protein construction then, if any two atoms are closer than allowed by van der Waals radii (8) a hard-sphere collision is deemed to occur. The most recently added residue would be removed and a new location chosen for it. In order to quickly test for collisions in three dimensions between new atoms and their nearest neighbors, a novel hierarchical algorithm, the binary-d tree (see Example 1), was employed and tested for finding nearby neighbors to a given point. A small one-time overhead of adding each new point to the tree, and removing them when forced to back up, ensures a rapid collision-test lookup later on.

Detailed Description Text (67):
The algorithm developed by the present inventors, based on a hierarchical data structure, partitions atoms in a molecular scene into nodes, as shown in FIG. 5, representing points in space relative to one another in a multidimensional tree, rather than enclosing them in some fixed volume as do oct-tree or grid methods. Unlike the oct-tree, this tree does not need to pre-declare a fixed volume in which the protein structure is to be made. The tree is referred to as a binary-d tree, since it is actually a binary tree in the first dimension, and conceptually could be used in other higher dimensions i.e. it could be extended into d binary trees at each node as shown in FIGS. 2A and 2B. The binary-d tree has nodes pointing to atomic coordinate locations, together with a set of three pairs of left/right child pointers. The difference between this tree and the oct-tree or k-d tree (19) is that it is comprised of points or vectors, rather than partitioned volumes or hyper-rectangles. It is also memory efficient, requiring only 2 pointers for each dimension (2d), as opposed to 2.sup.d pointers for oct-trees. In practice, a small one-time overhead is paid when adding each new atom's coordinates to the tree. Importantly, coordinates in this tree can be removed with only minor adjustments to the child nodes. This is useful when backtracking, as atoms can be removed without having to rebuild the entire tree.

Detailed Description Text (68):
Nearest neighbors in the tree are obtained by probing the tree with a multidimensional search using boundary coordinates that describe a three-dimensional rectangular probe volume shown in FIG. 3. The search returns a list of the atomic coordinates found in the probe volume. The method primarily distinguishes coordinates based on the first dimension, i.e. as a binary tree. The second and third set of left/right child pointers are rarely used. These higher dimensional left/right child pointers provide the key to higher-dimensional discrimination in the case where the first dimension is equivalent, as illustrated in the simple tree in FIGS. 2A and 2B, and add little or no overhead to the implementation or search method. Given the three decimal place precision found in the PDB, it is extremely likely (>99.99%) that any given 100-residue or more structure will have at least two atoms with identical x co-ordinates. The time-complexity of this algorithm was determined to be O(NlogN) with simple .alpha.-carbon kinetic random walks, prior to incorporation in the overall method.

Detailed Description Text (112):
Backtracking in a build-up method adds yet another dimension to the time complexity of the algorithm. The setting of 100 attempts was used to place a residue before backtracking, which seemed to work well in most cases. However, in some topologies, a combinatorial problem can arise if the only recourse for the build-up method is to backtrack many residues (e.g. backtracking N residues could require up to 100.sup.N failed placements). The kind of topology that causes this problem is a region of low conformational freedom (e.g. a helix) building up along a tube or channel with a dead-end, not uncommon in a folded structure. The only way out is to backtrack to the most recent turn.

Detailed Description Text (146):
so that for a protein with l=3.81 .ANG. and .theta..apprxeq.107.degree., R.sub.gyr.sup.2 =4.419N is expected. However, since a self-avoiding kinetic walk was used in this study, the walk is statistically biased, increasing the expected size. Also, since entire backbones and sidechains are being added on and tested for collisions, rather than just testing for

collisions between .alpha.-carbons, even larger values are expected. Using N=154 this gives an expected R.sub.gyr of 26.1 .ANG. for myoglobin. Hence the observed average of 27.3.+-.6.2 .ANG. when non-C.alpha. collisions and backbone optimization was ignored agrees with this. The distributions of R.sub.gyr shown in FIG. 12 illustrate the swelling of the distribution, as well as the mean, when full bump checking is used compared with the relatively narrow distribution achieved when only C.alpha. collisions can take place and all residues are treated as spheres of constant radius. It is also interesting to note that in both cases, (R.sub.gyr).sup.2 =5.86 was close to the expected limit of 6 as N approaches infinity. The theoretical equation for R.sub.gyr served as a lower bound for the observed values in the 995 randomly generated proteins as well, as can be seen in FIG. 14. The fitted exponent for the R.sub.gyr dependence on N of 0.57 is close to the value of 0.6 predicted by theory for a kinetic self-avoiding walk in 3 dimensions (Pietronero L. Phys.Rev.Lett. 1985;55:2025-2027).

Detailed Description Paragraph Table (1):
Listing 1 TYPE DECLARATIONS NODEPOINTER (->NODE) {notation -> indicates "points to"} NODE { array of real Dim[0. . .2] array of NODEPOINTER Way[0. . .5] } PROCEDURE PUTIN ARGUMENTS NODEPOINTER NewNode NODEPOINTER TreeNode integer dimension VARIABLES integer d integer branch BEGIN 1 branch = -1 2 d = dimension {the following determines the dimension for branching} 3 Do 4 IF (NewNode->Dim[d] > TreeNode->Dim[d]) 5 THEN 6 dimension = d 7 branch = (d * 2) + 1 8 ELSE 9 IF (NewNode->Dim[d] < TreeNode->Dim[d]) 10 THEN 11 dimension = d 12 branch = d * 2 13 d = d + 1 14 WHILE ((d <= 2) AND (branch = -1)) {based on the branch, traverse the tree or insert NewNode} 15 IF (branch >= 0) 16 THEN 17 IF (TreeNode->Way[branch] = NIL) 18 THEN TreeNode->Way [branch] = NewNode 19 ELSE 20 TreeNode = TreeNode->Way[branch] 21 PUTIN (NewNode, TreeNode, dimension) END GLOBAL VARIABLES NODE MinNode NODE MaxNode PROCEDURE XNeighbor ARGUMENTS NODEPOINTER TreeNode BEGIN 1 IF (TreeNode NOT = NIL) 2 IF (TreeNode->Dim[0]> MinNode->Dim(0)) 3 THEN XNeighbor(TreeNode->Way[0]) 4 YNeighbor (TreeNode) 5 IF(TreeNode->Dim[0] < MaxNode->Dim [0]) 6 THEN XNeighbor(TreeNode->Way[1]) END PROCEDURE YNeighbor ARGUMENTS NODEPOINTER TreeNode BEGIN 1 IF (TreeNode NOT = NIL) 2 IF (TreeNode->Dim[1] > MinNode->Dim[1]) 3 THEN YNeighbor (TreeNode->Way[2]) 4 ZNeighbor (TreeNode) 5 IF(TreeNode->Dim[1] < MaxNode->Dim[1]) 6 THEN YNeighbor(TreeNode->Way[3]) END PROCEDURE ZNeighbor ARGUMENTS NODEPOINTER TreeNode VARIABLES integer d integer found BEGIN 1 IF (TreeNode NOT = NIL) 2 IF (TreeNode->Dim[2] > MinNode->Dim [2]) 3 THEN ZNeighbor(TreeNode->Way[4]) 4 found = TRUE 5 FOR d = 0 TO 2 DO 6 IF ((TreeNode->Dim [d] > MaxNode->Dim[d]) OR (TreeNode->Dim[d] <= MinNode->Dim[d])) 7 THEN found = FALSE 6 IF (found = TRUE) 9 THEN ReportNeighbor(TreeNode) 10 IF(TreeNode->Dim[2] < MaxNode->Dim[2]) 11 THEN YNeighbor(TreeNode->Way[5]) END

Detailed Description Paragraph Table (2):
TABLE 1 Information contained in each trajectory distribution record of the database. Field Name Purpose Residue number Primary search key - index into database Amino acid One-letter amino acid code Dimension Size of trajectory distribution in both dimensions, normally 400, each residue can potentially have different resolutions of discretization Endianness Keeps track of what type of CPU the binary data was generated on to maintain platform independence of the file Compression type Indicates whether bzip2, RLE, or some other compression method was used on the trajectory distribution data Buffer size* Size of the compressed trajectory information (needed for decompression) Trajectory distribution data* Actual compressed binary data Integral* Total area under the trajectory distribution; units are arbitrary Peak value* Largest single value in the trajectory distribution - allows comparison of relative sizes of different trajectory distributions First non-zero row** First row in the trajectory distribution array which contains a non-zero number; the first row is row 1 Number of non-zero rows** The number of rows from the first non-zero row to the last one; rows of zeroes may appear amongst the non-zero ones, but this counts up to the last row with any non-zero data in it Number of elements <= 0*** Number of entries in the entire trajectory distribution array which are zero; gives an indication of shallowness Number of elements <= 5% peak*** Same as above, but includes up to 5% of the peak value Number or elements <= 10% peak*** Same as above, but includes up to 10% of the peak value Number of elements <= 15% peak*** Same as above, but includes up to 15% of the peak value Timeout Number of tries at each residue before backtracking Omega mean Average peptide dihedral angle between residue i-1 and i Omega standard deviation Standard deviation to use when choosing omega randomly Cis probability % probability that the bond between residue i and i + 1 will be cis (0 unless i + 1 is Pro) S-S probability % probability that residue is involved in a disulfide bridge (0 unless Cys) (for future use) Rotamer info Unused - possibly for storing favored rotamer index *has a corresponding field for when the residue is chosen to be cis, provided Cis probability is not zero. All other fields apply to both sets of trajectory distributions (cis and trans) where appropriate, unless

otherwise noted; cis trajectory distributions are used to place residue i + 1 when residue i is a cis-proline **always applies to trans trajectory distribution only; non-zero rows of cis trajectory distributions are stored explicitly ***always applies to most probable trajectory distribution, i.e. trans unless Cis probability > 0.5

Previous Doc Next Doc Go to Doc#

L10: Entry 6 of 8 File: USPT Oct 17, 2000

DOCUMENT-IDENTIFIER: US 6134541 A
TITLE: Searching multidimensional indexes using associated clustering and dimension reduction information

Parent Case Text (2):
The present invention is related to co-pending patent application Ser. No. 08/960,540, entitled "Multidimensional Data Clustering and Dimension Reduction for Indexing and Searching," by Castelli et al., ed of even date herewith, IBM Docket No. YO997170. This co-pending application and the present invention are commonly assigned to the International Business Machines Corporation, Armonk, N.Y. This co-pending application is hereby incorporated by reference in its entirety into the present application

Brief Summary Text (2):
The present invention is related to improved information retrieval systems. A particular aspect of the present invention is related to searching compact index representations of multidimensional data. A more particular aspect of the present invention is related to searching compact index representations of multidimensional data in database systems using associated clustering and dimension reduction information.

Brief Summary Text (10):
find record(s) that are within [a1 . . . a2], [b1 . . . b2], [z1 . . . z2] where a, b and z represent different dimensions (range search); and

Brief Summary Text (18):
Several well known spatial indexing techniques, such as R-trees can be used for range and nearest neighbor queries. Descriptions of R-trees can be found, for example, in "R-trees: A Dynamic index structure for spatial searching," by A. Guttman, ACM SIGMOD Conf. on Management of Data, Boston, Mass., June, 1994. The efficiency of these techniques, however, deteriorates rapidly as the number of dimensions of the feature space grows, since the search space becomes increasingly sparse. For instance, it is known that methods such as R-Trees are not useful when the number of dimensions is larger than 8, where the usefulness criterion is the time to complete a request compared to the time required by a brute force strategy the request by sequentially scanning every record in the database. The inefficiency of conventional indexing techniques in high dimensional spaces is a consequence of a well-known phenomenon called the "curse of dimensionality," which is described, for instance, in "From Statistics to Neural Networks," NATO ASI Series, vol. 136, Springer-Verlag, 1994, by V. Cherkassky, J. H. Friedman, and H. Wechsles. The relevant consequence of the curse of dimensionality is that clustering the index space into hypercubes is an inefficient method for feature spaces with a higher number of dimensions.

Brief Summary Text (19):
Because of the inefficiency associated with using existing spatial indexing techniques for indexing a high-dimensional feature space, techniques well known in the art exist to reduce the number of dimensions of a feature space. For example, the dimensionality can be reduced either by variable subset selection (also called feature selection) or by singular value decomposition followed by variable subset selection, as taught, for instance by C T. Chen, "Linear System Theory and Design", Holt, Rinehart and Winston, Appendix E, 1984. Variable subset selection is a well known and active field of study in statistics, and numerous methodologies have been proposed (see e.g., Shibata et al. "An Optimal Selection of Regression Variables," Biometrika vol. 68, No. 1, 1981, pp. 45-54. These methods are effective in an index generation system only if many of the variables (columns in the database) are highly correlated. This assumption is in general incorrect in real world databases.

Brief Summary Text (22):
In accordance with the aforementioned needs, the present invention is directed to an improved apparatus and method for efficiently performing exact and similarity searches on multidimensional data. One example of an application of the present invention is to multidimensional indexing. Multidimensional indexing is fundamental to spatial databases, which are widely applicable to: Geographic Information Systems (GIS); Online Analytical Processing (OLAP) for decision support using a large data warehouse; and products such as IBM's QBIC and IMAGEMINER for image mining of multimedia databases where high-dimensional feature vectors are extracted from image and video data.

Brief Summary Text (23):
The present invention has features for performing exact searches using one or more reduced dimensionality indexes to multidimensional data includes the steps of: associating specified data (such as a user-provided example or a template record) to a cluster, based on clustering information; reducing a dimensionality of the specified data, based on dimensionality reduction information for an associated reduced dimensionality cluster; and searching, based on the indexes and a reduced dimensionality specified data, for a reduced dimensionality version of the cluster matching the specified data. An example of the clustering information could be an identifier of a centroid of the cluster associated with a unique label.

Brief Summary Text (24):
If the index construction method used was recursive, then the dimension reduction and clustering information can also be used to locate the cluster where the target vector resides. An example of an exact search in a hierarchy of reduced dimensionality indexes includes the steps of: recursively applying the associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and searching, using low dimensionality reduction information for reduced dimensionality specified data, in response to the reducing step; and retrieving from the identified cluster, using the multidimensional index and the dimensionality reduction information for reduced dimensionality specified data, the records most similar to the specified data.

Brief Summary Text (26):
In yet another embodiment, wherein the specified data includes a search template, the dimension reduction includes the steps of: projecting the specified data onto a subspace for an associated cluster, based on dimensionality reduction information for the identified cluster; and generating dimensionality reduction information including an orthogonal complement for projected specified data, in response the projecting step. The projecting step can include producing a projected template and template dimensionality reduction information; the searching step, via the index, can be based on the projected template and the template dimensionality reduction information; and a k-nearest neighbor set of the k records most similar to the search template can be accordingly updated.

Brief Summary Text (27):
The present invention has also features for assessing if other clusters can contain elements that are closer to the specific data than the farthest of the k most similar element retrieved. As is known in the art, clustering information can be used to reconstruct the boundaries of the partitions, and these boundaries can be used to determine if a cluster can contain one of the k nearest neighbors. Those skilled in the art will appreciate that the cluster boundaries are a simple approximation to the structure of the cluster itself, namely, from the mathematical form of the dimensional indexes to the lowest level of the hierarchy, for a reduced dimensionality version of the cluster matching the specified data.

Brief Summary Text (29):
In a preferred embodiment, the dimension reduction step is a singular value decomposition, and the index is searched for a matching reduced dimensionality cluster, based on decomposed specified data. An example of the dimensionality reduction information is a transformation matrix (including eigenvalues and eigenvectors) generated by a singular value decomposition and selected eigenvalues of the transformation matrix.

Brief Summary Text (33):
In a preferred embodiment, the present invention is stored on a program storage device readable by a machine which uses one or more reduced dimensionality indexes to multidimensional data. The program storage device tangibly embodies it program of instructions in accordance with the

present invention and executable by the machine to perform method steps for an exact search for specified data using the one or more indexes, where the method steps include the steps of: associating specified data to a cluster, based on clustering information; reducing a dimensionality of the specified data, based on dimensionality reduction information for an associated reduced dimensionality cluster; and searching, based on the indexes and a reduced dimensionality specified data, for a reduced dimensionality version of the cluster matching the specified data.

Drawing Description Text (4):
FIG. 2 shows an example of the distribution of the data points and intuition for dimension reduction after clustering;

Detailed Description Text (6):
An example of an image mining application is QBIC, which is the integrated search facility in IBM's DB2 Image Extender. QBIC includes an image query engine (server), and a sample client consisting of an HTML graphical user interface and related common gateway interface (CGI) scripts that together form the basis of a complete application. Both the server and the client are extensible so that a user can develop an application-specific image matching function and add it to QBIC. The image search server allows queries of large image databases based on visual image content. It features:

Detailed Description Text (14):
where `N` is the number of dimensions of the vector that is used for indexing.

Detailed Description Text (16):
(1) Exact queries: where a vector is specified and the records or multimedia data that match the vector will be retrieved;

Detailed Description Text (17):
(2) Range queries: where the lower and upper limit of each dimension of the vector is specified.

Detailed Description Text (20):
Note that it is not necessary for all of the dimensions i to participate in the computation of either a range or nearest neighbor query. In both cases, a subset of the dimensions can be specified to retrieve the results.

Detailed Description Text (21):
FIG. 2 shows an example of the distribution of the vectors in a multidimensional space. As depicted, a total of three dimensions are required to represent the entire space. However, only two dimensions are required to represent each individual cluster, as cluster 201, 202, and 203 are located on the x-y, y-z, and z-x planes, respectively. Thus, it can be concluded that dimension reduction can be achieved through proper clustering of the data. The same dimensional reduction cannot be achieved by singular value decomposition alone, which can only re-orient the feature space so that the axis in the space coincides with the dominant dimensions (three in this example).

Detailed Description Text (22):
Eliminating one or more dimensions of a vector is equivalent to projecting the original points into a subspace. Equation (2) shows that only those dimensions where the individual elements in the vector are different, need to be computed. As a result, the projection of the vector into a subspace does not affect the computation of the distance, providing those elements that are eliminated do not vary in the original space.

Detailed Description Text (25):
In the following, a methodology will be derived to estimate the maximum error that can result from projecting vectors into a subspace. The process starts by determining the bound of the maximum error. Denoting the centroid of a cluster as Vc, which is defined as ##EQU1## where N is the total number of vectors in the cluster, which consists of vectors {V1, . . . , VN}. After the cluster is projected into a k dimensional subspace, where without loss of generality the last (n-k) dimensions are eliminated, an error is introduced to the distance between any two vectors in the subspace as compared to the original space. The error term is ##EQU2## The following inequality immediately hold: ##EQU3## Equation (5) shows that the maximum error

incurred by computing the distance in the projected subspace is bounded.

Detailed Description Text (34):
In step 709, the transformation logic takes the data cluster (702) and the transformation matrix (703) as input; and applies a transformation specified by the transformation matrix (703) to the elements of the data cluster (702) and produces a transformed data cluster (710). In step 711, the selected eigenvalues (708) and the transformed data cluster (710) are used to produce the reduced dimensionality data cluster (712). In a preferred embodiment, the dimensionality reduction is accomplished by retaining the smallest number of dimensions such that the set of corresponding eigenvalues account for at least a fixed percentage of the total variance, where for instance the fixed percentage can be taken to be equal to 95%.

Detailed Description Text (38):
Note that as the number of returned results n increases, the precision decreases while the recall increases. In general, the trends of precision and recall are not monotonic. Since E(c) depends on n, an efficiency vs. recall curve is often plotted as a parametric function of n. In a preferred embodiment, a requester specifies the desired precision of the search and a lower bound on the allowed recall. Then the dimensionality reduction logic performs the dimensionality reduction based on precision and recall as follows: after ordering the eigenvalues in decreasing order, the dimensionality reduction logic (step 606, FIG. 6) removes the dimension corresponding to the smallest eigenvalue, and estimates the resulting precision vs. recall function based on a test set of samples selected at random from the original training set or provided by the user. From the precision vs. recall function, the dimensionality reduction logic derives a maximum value of precision n.sub.max for which the desired recall is attained. Then the dimensionality reduction logic iterates the same procedure by removing the dimension corresponding to the next smallest eigenvalue, and computes the corresponding precision for which the desired recall is attained. The iterative procedure is terminated when the computed precision is below the threshold value specified by the user, and the dimensionality reduction logic retains only the dimensions retained at the iteration immediately preceding the one where the termination condition occurs.

Detailed Description Text (39):
In another embodiment of the present invention, the requester specifies only a value of desired recall, and the dimensionality reduction logic estimates the cost of increasing the precision to attain the desired recall. This cost has two components: one that decreases with the number of dimensions, since computing distances and searching for nearest neighbors is more efficient in lower dimensionality spaces; and an increasing component due to the fact that the number of retrieved results must grow as the number of retained dimensions is reduced to insure the desired value of recall. Retrieving a larger number n of nearest neighbors is more expensive even when using efficient methods, since the portion of the search space that must be analyzed grows with the number of desired results. Then the dimensionality reduction logic finds by exhaustive search, the number of dimensions to retain that minimizes the cost of the search for the user-specified value of the recall.

Detailed Description Text (40):
The clustering and singular value decomposition can be applied to the vectors recursively (step 601-611) until a terminating condition (step 609) is reached. One such terminating condition can be that the dimension of each cluster can no longer be reduced as described herein. Optionally, more conventional spatial indexing techniques such as the R-tree can then be applied to each cluster. These techniques are much more efficient for those clusters whose dimension have been minimized. This would thus complete the entire index generation process for a set of high dimensional vectors.

Detailed Description Text (42):
FIG. 8 shows an example of a logic flow for an exact search process based on a searchable index (108 or 612) generated according to the present invention. In this example, the index is generated without recursive application of the clustering and singular value decomposition. An exact search is the process of retrieving a record or records that exactly match a search query, such as a search template. As depicted, in step 802 a query including specified data such as a search template (801) is received by the multidimensional index engine (107) (also called cluster search logic). In step 802, the clustering information (604) produced in step 601 of FIG. 6, is used to identify the cluster to which the search template belongs. In step 803, the dimensionality reduction information (607) produced in step 606 of FIG. 6, is used to

project the input template onto the subspace associated with the cluster identified in step 802, and produce a projected template (804). In step 805, an intra-cluster search logic uses the searchable index (612) generated in step 610 of FIG. 6 to search for the projected template. Note that the simplest search mechanism within each cluster is to conduct a linear scan (or linear search) if no spatial indexing structure can be utilized. On most occasions, spatial indexing structures such as R-trees can offer better efficiency (as compared to linear scan) when the dimension of the cluster is relatively small (smaller than 10 in most cases).

Detailed Description Text (43):
FIG. 9 shows another example of a flow chart of an exact search process based on a searchable multidimensional index (108 or 612) generated according to the present invention. Here, the index (108 or 612) was generated with a recursive application of the clustering and dimensionality reduction logic. An exact search is the process of retrieving the record or records that exactly match the search template. As depicted, a query including specified data such as a search template (901) is used as input to the cluster search logic, in step 902, which is analogous to the cluster search logic in step 802 of FIG. 8. In step 902, clustering information (604) produced in step (601) of FIG. 6 is used to identify the cluster to which the search template (901) belongs. In step 903, (analogous to step 803 of FIG. 8) the dimensionality reduction information (607), produced in step 606 of FIG. 6, is used to project the input template onto the subspace associated with the cluster identified in step 902, and produce a projected template (904). In step 905, it is determined whether the current cluster is terminal, that is, if no further recursive clustering and singular value decomposition steps were applied to this cluster during the multidimensional index construction process. If the cluster is not terminal, in step 907 the search template (901) is replaced by the projected template (904), and the process returns to step 902. In step 906, if the cluster is terminal, the intra-cluster search logic uses the searchable index to search for the projected template. As noted, the simplest intra-cluster search mechanism is to conduct a linear scan (or linear search), if no spatial indexing structure can be utilized. On most occasions, spatial indexing structures such as R-trees can offer better efficiency (as compared to linear scan) when the dimension of the cluster is relatively small (smaller than 10 in most cases).

Detailed Description Text (46):
FIG. 10 shows an example of a flow chart of a k-nearest neighbor search process based on an index (612) generated according to the present invention. In this example, the index is generated without recursive application of the clustering and singular value decomposition. The k-nearest neighbor search returns the k closest entries in the database which match the query. The number (nr) of desired matches, k (1000) is used in step 1002 to initialize the k-nearest neighbor set (1009) so that it contains at most k elements and so that it: is empty before the next step begins. In step 1003, the cluster search logic receives the query, for example a the search template (1001) and determines which cluster the search template belongs to, using the clustering information (604) produced in step 601 of FIG. 6. In step 1004, template is then projected onto the subspace associated with the cluster it belongs to, using the dimensionality reduction information (607). The projection step 1004 produces a projected template (1006) and dimensionality reduction information (1005), that includes an orthogonal complement of the projected template (1006) (defined as the vector difference of the search template (1001) and the projected template (1006)), and the Euclidean length of the orthogonal complement. The dimensionality reduction information (1005) and the projected template (1006) can be used by the intra-cluster search logic, in step 1007, which updates the k-nearest neighbor set (1009), using the multidimensional index. Examples of the intra-cluster search logic adaptable according to the present invention include any of the nearest neighbor search methods known in the art; see e.g., "Nearest Neighbor Pattern Classification Techniques," Belur V. Desarathy, (editor), IEEE Computer Society (1991). An example of an intra-cluster search logic (step 1007) having features of the present invention includes the steps of: first, the squared distance between the projected template (1006) and the members of the cluster in the vector space with reduced dimensionality is computed; the result is added to the squared distance between the search template (1001) and the subspace of the cluster; the final result is defined as the "sum" of the squared lengths of the orthogonal complements, (computed in step (1004) which is part of the dimensionality reduction information (1005):

Detailed Description Text (50):
FIG. 11 shows an example of a flow chart of a k-nearest neighbor search process based on a search index (612) generated according to the present invention. Here, the index is generated with a recursive application of the clustering and singular value decomposition. The k-nearest

neighbor search returns the closest k entries in the database, to specified data in the form of a search template. As depicted, in step 1102, the k-nearest neighbor set is initialized to empty and the number of desired matches, k (1100) is used to initialize the k-nearest neighbor set (1111) so that it can contain at most k elements. In step 1103, the cluster search logic takes the search template (1101) as input and associates the search template to a corresponding cluster, using the clustering information (604) produced in step 601 of FIG. 6. In step 1104, the template (1101) is projected onto a subspace associated with the cluster identified in step

Detailed Description Text (58):
FIG. 14 shows an example of a complex surface (1401) in a 3-dimensional space and two successive approximations (1402, 1403) based on a 3-dimensional quad tree, as taught in the art by Samet, H. in "Region Representation Quadtree from Boundary Codes" Comm. ACM 23,3, pp. 163-170 (March 1980). The first approximation (1402) is a minimal bounding box. The second approximation (1403) is the second step of a quad tree generation, where the bounding box has been divided into 8 hyper rectangles by splitting the bounding box at the midpoint of each dimension, and retaining only those hyper rectangles that intersect the surface.

Detailed Description Text (59):
In a preferred embodiment, the hierarchy of approximations is generated as a k-dimensional quad tree. An example of a method having features of the present invention for generating the hierarchy of approximations includes the steps of: generating the cluster boundaries, which correspond to a zeroth-order approximation to the geometry of the clusters; approximating the convex hull of each of the clusters by means of a minimum bounding box, thus generating a first-order approximation to the geometry of each cluster, partitioning the bounding box into 2.sup.k hyper rectangles, by cutting it at the half point of each dimension; retaining only those hyper rectangles that contain points, thus generating a second-order approximation to the geometry of the cluster; and repeating the last two steps for each of the retained hyper rectangles to generate successively the third-, fourth-, . . . , n-th approximation to the geometry of the cluster.

CLAIMS:

1. In a system including one or more reduced dimensionality indexes to multidimensional data, a method for performing an exact search for specified data using the one or more indexes, the method comprising the following steps in the sequence set forth:

associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps unitl a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

searching, using low dimensional indexes to said lowest level and a reduced dimensionality specified data, for cluster elements of the reduced dimensionality version of the cluster matching the specified data.

2. A method for performing an exact search for specified data, comprising the following steps in the sequence set forth:

associating specified data to a data cluster based on clustering information, said data cluster being a partition from an original data input set;

reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

linearly scanning, based on a reduced dimensionality specified data, for the reduced

dimensionality version of the cluster matching the specified data.

3. The method of claim 1, wherein said reducing step comprises a singular value decomposition, said searching further comprising the step of:

searching an index for a matching reduced dimensionality cluster, based on decomposed specified data.

7. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for performing an exact search for specified data, said method comprising:

associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

after said associating, reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

linearly scanning, based on a reduced dimensionality specified data, for the reduced dimensionality version of the cluster matching the specified data.

21. A program storage device readable by a machine which includes one or more reduced dimensionality indexes to multidimensional data, the program storage device tangibly embodying a program of instructions executable by the machine to perform method steps for performing an exact search for specified data using the one or more indexes, said method steps comprising:

associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

searching, using low dimensional indexes to said lowest level and a reduced dimensionality specified data, for cluster elements of the reduced dimensionality version of the cluster matching the specified data.

23. The program storage device of claim 21, wherein said reducing step comprises a singular value decomposition, said searching further comprising the step of:

searching an index for a matching reduced dimensionality cluster, based on decomposed specified data.

41. A program storage device readable by a machine which includes one or more reduced dimensionality indexes to multidimensional data, the program storage device tangibly embodying a program of instructions executable by the machine to perform method steps for searching for k records most similar to specified data, using the one or more indexes, said method steps comprising:

identifying the specified data with a cluster based on clustering information, said cluster being a partition of an original data input set;

after the identifying step, reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of an identified cluster;

recursively applying said identifying and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached;

generating dimensionality reduction information for reduced dimensionality specified data, in response to said reducing; and

linearly scanning, based on the reduced dimensionality specified data, for the reduced dimensionality version of the cluster matching the specified data.

42. A computer program product comprising:

a computer usable medium having computer readable program code means embodied therein for performing an exact search for a search template using one or more multidimensional indexes, the computer readable program code means in said computer program product comprising:

computer readable program code cluster search means for causing a computer to effect associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

computer readable program code template projection means for causing a computer to effect reducing a dimensionality of the search template, after said associating, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

computer readable program code means for causing a computer to effect recursively applying said cluster search means and template projection means until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

computer readable program code intra-cluster search means for causing a computer to effect searching, using low dimensional indexes to said lowest level and a reduced dimensionality search template, for cluster elements of the reduced dimensionality version of the cluster matching the search template.

Previous Doc Next Doc Go to Doc#

L10: Entry 6 of 8 File: USPT Oct 17, 2000

DOCUMENT-IDENTIFIER: US 6134541 A
TITLE: Searching multidimensional indexes using associated clustering and dimension reduction information

Parent Case Text (2):
The present invention is related to co-pending patent application Ser. No. 08/960,540, entitled "Multidimensional Data Clustering and Dimension Reduction for Indexing and Searching," by Castelli et al., ed of even date herewith, IBM Docket No. YO997170. This co-pending application and the present invention are commonly assigned to the International Business Machines Corporation, Armonk, N.Y. This co-pending application is hereby incorporated by reference in its entirety into the present application

Brief Summary Text (2):
The present invention is related to improved information retrieval systems. A particular aspect of the present invention is related to searching compact index representations of multidimensional data. A more particular aspect of the present invention is related to searching compact index representations of multidimensional data in database systems using associated clustering and dimension reduction information.

Brief Summary Text (10):
find record(s) that are within [a1 . . . a2], [b1 . . . b2], [z1 . . . z2] where a, b and z represent different dimensions (range search); and

Brief Summary Text (18):
Several well known spatial indexing techniques, such as R-trees can be used for range and nearest neighbor queries. Descriptions of R-trees can be found, for example, in "R-trees: A Dynamic index structure for spatial searching," by A. Guttman, ACM SIGMOD Conf. on Management of Data, Boston, Mass., June, 1994. The efficiency of these techniques, however, deteriorates rapidly as the number of dimensions of the feature space grows, since the search space becomes increasingly sparse. For instance, it is known that methods such as R-Trees are not useful when the number of dimensions is larger than 8, where the usefulness criterion is the time to complete a request compared to the time required by a brute force strategy the request by sequentially scanning every record in the database. The inefficiency of conventional indexing techniques in high dimensional spaces is a consequence of a well-known phenomenon called the "curse of dimensionality," which is described, for instance, in "From Statistics to Neural Networks," NATO ASI Series, vol. 136, Springer-Verlag, 1994, by V. Cherkassky, J. H. Friedman, and H. Wechsles. The relevant consequence of the curse of dimensionality is that clustering the index space into hypercubes is an inefficient method for feature spaces with a higher number of dimensions.

Brief Summary Text (19):
Because of the inefficiency associated with using existing spatial indexing techniques for indexing a high-dimensional feature space, techniques well known in the art exist to reduce the number of dimensions of a feature space. For example, the dimensionality can be reduced either by variable subset selection (also called feature selection) or by singular value decomposition followed by variable subset selection, as taught, for instance by C T. Chen, "Linear System Theory and Design", Holt, Rinehart and Winston, Appendix E, 1984. Variable subset selection is a well known and active field of study in statistics, and numerous methodologies have been proposed (see e.g., Shibata et al. "An Optimal Selection of Regression Variables," Biometrika vol. 68, No. 1, 1981, pp. 45-54. These methods are effective in an index generation system only if many of the variables (columns in the database) are highly correlated. This assumption is in general incorrect in real world databases.

Brief Summary Text (22):
In accordance with the aforementioned needs, the present invention is directed to an improved apparatus and method for efficiently performing exact and similarity searches on multidimensional data. One example of an application of the present invention is to multidimensional indexing. Multidimensional indexing is fundamental to spatial databases, which are widely applicable to: Geographic Information Systems (GIS); Online Analytical Processing (OLAP) for decision support using a large data warehouse; and products such as IBM's QBIC and IMAGEMINER for image mining of multimedia databases where high-dimensional feature vectors are extracted from image and video data.

Brief Summary Text (23):
The present invention has features for performing exact searches using one or more reduced dimensionality indexes to multidimensional data includes the steps of: associating specified data (such as a user-provided example or a template record) to a cluster, based on clustering information; reducing a dimensionality of the specified data, based on dimensionality reduction information for an associated reduced dimensionality cluster; and searching, based on the indexes and a reduced dimensionality specified data, for a reduced dimensionality version of the cluster matching the specified data. An example of the clustering information could be an identifier of a centroid of the cluster associated with a unique label.

Brief Summary Text (24):
If the index construction method used was recursive, then the dimension reduction and clustering information can also be used to locate the cluster where the target vector resides. An example of an exact search in a hierarchy of reduced dimensionality indexes includes the steps of: recursively applying the associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and searching, using low dimensionality reduction information for reduced dimensionality specified data, in response to the reducing step; and retrieving from the identified cluster, using the multidimensional index and the dimensionality reduction information for reduced dimensionality specified data, the records most similar to the specified data.

Brief Summary Text (26):
In yet another embodiment, wherein the specified data includes a search template, the dimension reduction includes the steps of: projecting the specified data onto a subspace for an associated cluster, based on dimensionality reduction information for the identified cluster; and generating dimensionality reduction information including an orthogonal complement for projected specified data, in response the projecting step. The projecting step can include producing a projected template and template dimensionality reduction information; the searching step, via the index, can be based on the projected template and the template dimensionality reduction information; and a k-nearest neighbor set of the k records most similar to the search template can be accordingly updated.

Brief Summary Text (27):
The present invention has also features for assessing if other clusters can contain elements that are closer to the specific data than the farthest of the k most similar element retrieved. As is known in the art, clustering information can be used to reconstruct the boundaries of the partitions, and these boundaries can be used to determine if a cluster can contain one of the k nearest neighbors. Those skilled in the art will appreciate that the cluster boundaries are a simple approximation to the structure of the cluster itself, namely, from the mathematical form of the dimensional indexes to the lowest level of the hierarchy, for a reduced dimensionality version of the cluster matching the specified data.

Brief Summary Text (29):
In a preferred embodiment, the dimension reduction step is a singular value decomposition, and the index is searched for a matching reduced dimensionality cluster, based on decomposed specified data. An example of the dimensionality reduction information is a transformation matrix (including eigenvalues and eigenvectors) generated by a singular value decomposition and selected eigenvalues of the transformation matrix.

Brief Summary Text (33):
In a preferred embodiment, the present invention is stored on a program storage device readable by a machine which uses one or more reduced dimensionality indexes to multidimensional data. The program storage device tangibly embodies it program of instructions in accordance with the

present invention and executable by the machine to perform method steps for an exact search for specified data using the one or more indexes, where the method steps include the steps of: associating specified data to a cluster, based on clustering information; reducing a dimensionality of the specified data, based on dimensionality reduction information for an associated reduced dimensionality cluster; and searching, based on the indexes and a reduced dimensionality specified data, for a reduced dimensionality version of the cluster matching the specified data.

Drawing Description Text (4):
FIG. 2 shows an example of the distribution of the data points and intuition for dimension reduction after clustering;

Detailed Description Text (6):
An example of an image mining application is QBIC, which is the integrated search facility in IBM's DB2 Image Extender. QBIC includes an image query engine (server), and a sample client consisting of an HTML graphical user interface and related common gateway interface (CGI) scripts that together form the basis of a complete application. Both the server and the client are extensible so that a user can develop an application-specific image matching function and add it to QBIC. The image search server allows queries of large image databases based on visual image content. It features:

Detailed Description Text (14):
where `N` is the number of dimensions of the vector that is used for indexing.

Detailed Description Text (16):
(1) Exact queries: where a vector is specified and the records or multimedia data that match the vector will be retrieved;

Detailed Description Text (17):
(2) Range queries: where the lower and upper limit of each dimension of the vector is specified.

Detailed Description Text (20):
Note that it is not necessary for all of the dimensions i to participate in the computation of either a range or nearest neighbor query. In both cases, a subset of the dimensions can be specified to retrieve the results.

Detailed Description Text (21):
FIG. 2 shows an example of the distribution of the vectors in a multidimensional space. As depicted, a total of three dimensions are required to represent the entire space. However, only two dimensions are required to represent each individual cluster, as cluster 201, 202, and 203 are located on the x-y, y-z, and z-x planes, respectively. Thus, it can be concluded that dimension reduction can be achieved through proper clustering of the data. The same dimensional reduction cannot be achieved by singular value decomposition alone, which can only re-orient the feature space so that the axis in the space coincides with the dominant dimensions (three in this example).

Detailed Description Text (22):
Eliminating one or more dimensions of a vector is equivalent to projecting the original points into a subspace. Equation (2) shows that only those dimensions where the individual elements in the vector are different, need to be computed. As a result, the projection of the vector into a subspace does not affect the computation of the distance, providing those elements that are eliminated do not vary in the original space.

Detailed Description Text (25):
In the following, a methodology will be derived to estimate the maximum error that can result from projecting vectors into a subspace. The process starts by determining the bound of the maximum error. Denoting the centroid of a cluster as Vc, which is defined as ##EQU1## where N is the total number of vectors in the cluster, which consists of vectors {V1, . . . , VN}. After the cluster is projected into a k dimensional subspace, where without loss of generality the last (n-k) dimensions are eliminated, an error is introduced to the distance between any two vectors in the subspace as compared to the original space. The error term is ##EQU2## The following inequality immediately hold: ##EQU3## Equation (5) shows that the maximum error

incurred by computing the distance in the projected subspace is bounded.

Detailed Description Text (34):
In step 709, the transformation logic takes the data cluster (702) and the transformation matrix (703) as input; and applies a transformation specified by the transformation matrix (703) to the elements of the data cluster (702) and produces a transformed data cluster (710). In step 711, the selected eigenvalues (708) and the transformed data cluster (710) are used to produce the reduced dimensionality data cluster (712). In a preferred embodiment, the dimensionality reduction is accomplished by retaining the smallest number of dimensions such that the set of corresponding eigenvalues account for at least a fixed percentage of the total variance, where for instance the fixed percentage can be taken to be equal to 95%.

Detailed Description Text (38):
Note that as the number of returned results n increases, the precision decreases while the recall increases. In general, the trends of precision and recall are not monotonic. Since E(c) depends on n, an efficiency vs. recall curve is often plotted as a parametric function of n. In a preferred embodiment, a requester specifies the desired precision of the search and a lower bound on the allowed recall. Then the dimensionality reduction logic performs the dimensionality reduction based on precision and recall as follows: after ordering the eigenvalues in decreasing order, the dimensionality reduction logic (step 606, FIG. 6) removes the dimension corresponding to the smallest eigenvalue, and estimates the resulting precision vs. recall function based on a test set of samples selected at random from the original training set or provided by the user. From the precision vs. recall function, the dimensionality reduction logic derives a maximum value of precision n.sub.max for which the desired recall is attained. Then the dimensionality reduction logic iterates the same procedure by removing the dimension corresponding to the next smallest eigenvalue, and computes the corresponding precision for which the desired recall is attained. The iterative procedure is terminated when the computed precision is below the threshold value specified by the user, and the dimensionality reduction logic retains only the dimensions retained at the iteration immediately preceding the one where the termination condition occurs.

Detailed Description Text (39):
In another embodiment of the present invention, the requester specifies only a value of desired recall, and the dimensionality reduction logic estimates the cost of increasing the precision to attain the desired recall. This cost has two components: one that decreases with the number of dimensions, since computing distances and searching for nearest neighbors is more efficient in lower dimensionality spaces; and an increasing component due to the fact that the number of retrieved results must grow as the number of retained dimensions is reduced to insure the desired value of recall. Retrieving a larger number n of nearest neighbors is more expensive even when using efficient methods, since the portion of the search space that must be analyzed grows with the number of desired results. Then the dimensionality reduction logic finds by exhaustive search, the number of dimensions to retain that minimizes the cost of the search for the user-specified value of the recall.

Detailed Description Text (40):
The clustering and singular value decomposition can be applied to the vectors recursively (step 601-611) until a terminating condition (step 609) is reached. One such terminating condition can be that the dimension of each cluster can no longer be reduced as described herein. Optionally, more conventional spatial indexing techniques such as the R-tree can then be applied to each cluster. These techniques are much more efficient for those clusters whose dimension have been minimized. This would thus complete the entire index generation process for a set of high dimensional vectors.

Detailed Description Text (42):
FIG. 8 shows an example of a logic flow for an exact search process based on a searchable index (108 or 612) generated according to the present invention. In this example, the index is generated without recursive application of the clustering and singular value decomposition. An exact search is the process of retrieving a record or records that exactly match a search query, such as a search template. As depicted, in step 802 a query including specified data such as a search template (801) is received by the multidimensional index engine (107) (also called cluster search logic). In step 802, the clustering information (604) produced in step 601 of FIG. 6, is used to identify the cluster to which the search template belongs. In step 803, the dimensionality reduction information (607) produced in step 606 of FIG. 6, is used to

project the input template onto the subspace associated with the cluster identified in step 802, and produce a projected template (804). In step 805, an intra-cluster search logic uses the searchable index (612) generated in step 610 of FIG. 6 to search for the projected template. Note that the simplest search mechanism within each cluster is to conduct a linear scan (or linear search) if no spatial indexing structure can be utilized. On most occasions, spatial indexing structures such as R-trees can offer better efficiency (as compared to linear scan) when the dimension of the cluster is relatively small (smaller than 10 in most cases).

Detailed Description Text (43):
FIG. 9 shows another example of a flow chart of an exact search process based on a searchable multidimensional index (108 or 612) generated according to the present invention. Here, the index (108 or 612) was generated with a recursive application of the clustering and dimensionality reduction logic. An exact search is the process of retrieving the record or records that exactly match the search template. As depicted, a query including specified data such as a search template (901) is used as input to the cluster search logic, in step 902, which is analogous to the cluster search logic in step 802 of FIG. 8. In step 902, clustering information (604) produced in step (601) of FIG. 6 is used to identify the cluster to which the search template (901) belongs. In step 903, (analogous to step 803 of FIG. 8) the dimensionality reduction information (607), produced in step 606 of FIG. 6, is used to project the input template onto the subspace associated with the cluster identified in step 902, and produce a projected template (904). In step 905, it is determined whether the current cluster is terminal, that is, if no further recursive clustering and singular value decomposition steps were applied to this cluster during the multidimensional index construction process. If the cluster is not terminal, in step 907 the search template (901) is replaced by the projected template (904), and the process returns to step 902. In step 906, if the cluster is terminal, the intra-cluster search logic uses the searchable index to search for the projected template. As noted, the simplest intra-cluster search mechanism is to conduct a linear scan (or linear search), if no spatial indexing structure can be utilized. On most occasions, spatial indexing structures such as R-trees can offer better efficiency (as compared to linear scan) when the dimension of the cluster is relatively small (smaller than 10 in most cases).

Detailed Description Text (46):
FIG. 10 shows an example of a flow chart of a k-nearest neighbor search process based on an index (612) generated according to the present invention. In this example, the index is generated without recursive application of the clustering and singular value decomposition. The k-nearest neighbor search returns the k closest entries in the database which match the query. The number (nr) of desired matches, k (1000) is used in step 1002 to initialize the k-nearest neighbor set (1009) so that it contains at most k elements and so that it: is empty before the next step begins. In step 1003, the cluster search logic receives the query, for example a the search template (1001) and determines which cluster the search template belongs to, using the clustering information (604) produced in step 601 of FIG. 6. In step 1004, template is then projected onto the subspace associated with the cluster it belongs to, using the dimensionality reduction information (607). The projection step 1004 produces a projected template (1006) and dimensionality reduction information (1005), that includes an orthogonal complement of the projected template (1006) (defined as the vector difference of the search template (1001) and the projected template (1006)), and the Euclidean length of the orthogonal complement. The dimensionality reduction information (1005) and the projected template (1006) can be used by the intra-cluster search logic, in step 1007, which updates the k-nearest neighbor set (1009), using the multidimensional index. Examples of the intra-cluster search logic adaptable according to the present invention include any of the nearest neighbor search methods known in the art; see e.g., "Nearest Neighbor Pattern Classification Techniques," Belur V. Desarathy, (editor), IEEE Computer Society (1991). An example of an intra-cluster search logic (step 1007) having features of the present invention includes the steps of: first, the squared distance between the projected template (1006) and the members of the cluster in the vector space with reduced dimensionality is computed; the result is added to the squared distance between the search template (1001) and the subspace of the cluster; the final result is defined as the "sum" of the squared lengths of the orthogonal complements, (computed in step (1004) which is part of the dimensionality reduction information (1005):

Detailed Description Text (50):
FIG. 11 shows an example of a flow chart of a k-nearest neighbor search process based on a search index (612) generated according to the present invention. Here, the index is generated with a recursive application of the clustering and singular value decomposition. The k-nearest

neighbor search returns the closest k entries in the database, to specified data in the form of a search template. As depicted, in step 1102, the k-nearest neighbor set is initialized to empty and the number of desired matches, k (1100) is used to initialize the k-nearest neighbor set (1111) so that it can contain at most k elements. In step 1103, the cluster search logic takes the search template (1101) as input and associates the search template to a corresponding cluster, using the clustering information (604) produced in step 601 of FIG. 6. In step 1104, the template (1101) is projected onto a subspace associated with the cluster identified in step

Detailed Description Text (58):
FIG. 14 shows an example of a complex surface (1401) in a 3-dimensional space and two successive approximations (1402; 1403) based on a 3-dimensional quad tree, as taught in the art by Samet, H. in "Region Representation Quadtree from Boundary Codes" Comm. ACM 23,3, pp. 163-170 (March 1980). The first approximation (1402) is a minimal bounding box. The second approximation (1403) is the second step of a quad tree generation, where the bounding box has been divided into 8 hyper rectangles by splitting the bounding box at the midpoint of each dimension, and retaining only those hyper rectangles that intersect the surface.

Detailed Description Text (59):
In a preferred embodiment, the hierarchy of approximations is generated as a k-dimensional quad tree. An example of a method having features of the present invention for generating the hierarchy of approximations includes the steps of: generating the cluster boundaries, which correspond to a zeroth-order approximation to the geometry of the clusters; approximating the convex hull of each of the clusters by means of a minimum bounding box, thus generating a first-order approximation to the geometry of each cluster, partitioning the bounding box into 2.sup.k hyper rectangles, by cutting it at the half point of each dimension; retaining only those hyper rectangles that contain points, thus generating a second-order approximation to the geometry of the cluster; and repeating the last two steps for each of the retained hyper rectangles to generate successively the third-, fourth-, . . . , n-th approximation to the geometry of the cluster.

CLAIMS:

1. In a system including one or more reduced dimensionality indexes to multidimensional data, a method for performing an exact search for specified data using the one or more indexes, the method comprising the following steps in the sequence set forth:

associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps unitl a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

searching, using low dimensional indexes to said lowest level and a reduced dimensionality specified data, for cluster elements of the reduced dimensionality version of the cluster matching the specified data.

2. A method for performing an exact search for specified data, comprising the following steps in the sequence set forth:

associating specified data to a data cluster based on clustering information, said data cluster being a partition from an original data input set;

reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

linearly scanning, based on a reduced dimensionality specified data, for the reduced

dimensionality version of the cluster matching the specified data.

3. The method of claim 1, wherein said reducing step comprises a singular value decomposition, said searching further comprising the step of:

searching an index for a matching reduced dimensionality cluster, based on decomposed specified data.

7. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for performing an exact search for specified data, said method comprising:

associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

after said associating, reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

linearly scanning, based on a reduced dimensionality specified data, for the reduced dimensionality version of the cluster matching the specified data.

21. A program storage device readable by a machine which includes one or more reduced dimensionality indexes to multidimensional data, the program storage device tangibly embodying a program of instructions executable by the machine to perform method steps for performing an exact search for specified data using the one or more indexes, said method steps comprising:

associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

recursively applying said associating and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

searching, using low dimensional indexes to said lowest level and a reduced dimensionality specified data, for cluster elements of the reduced dimensionality version of the cluster matching the specified data.

23. The program storage device of claim 21, wherein said reducing step comprises a singular value decomposition, said searching further comprising the step of:

searching an index for a matching reduced dimensionality cluster, based on decomposed specified data.

41. A program storage device readable by a machine which includes one or more reduced dimensionality indexes to multidimensional data, the program storage device tangibly embodying a program of instructions executable by the machine to perform method steps for searching for k records most similar to specified data, using the one or more indexes, said method steps comprising:

identifying the specified data with a cluster based on clustering information, said cluster being a partition of an original data input set;

after the identifying step, reducing a dimensionality of the specified data, based on dimensionality reduction information for a reduced dimensionality version of an identified cluster;

recursively applying said identifying and reducing steps until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached;

generating dimensionality reduction information for reduced dimensionality specified data, in response to said reducing; and

linearly scanning, based on the reduced dimensionality specified data, for the reduced dimensionality version of the cluster matching the specified data.

42. A computer program product comprising:

a computer usable medium having computer readable program code means embodied therein for performing an exact search for a search template using one or more multidimensional indexes, the computer readable program code means in said computer program product comprising:

computer readable program code cluster search means for causing a computer to effect associating specified data to a data cluster based on clustering information, said data cluster being a partition of an original data input set;

computer readable program code template projection means for causing a computer to effect reducing a dimensionality of the search template, after said associating, based on dimensionality reduction information for a reduced dimensionality version of the cluster;

computer readable program code means for causing a computer to effect recursively applying said cluster search means and template projection means until a corresponding lowest level of a hierarchy of reduced dimensionality clusters has been reached; and

computer readable program code intra-cluster search means for causing a computer to effect searching, using low dimensional indexes to said lowest level and a reduced dimensionality search template, for cluster elements of the reduced dimensionality version of the cluster matching the search template.

Previous Doc Next Doc Go to Doc#

L7: Entry 2 of 2 File: USPT Jul 14, 1998

DOCUMENT-IDENTIFIER: US 5781906 A
TITLE: System and method for construction of a data structure for indexing multidimensional objects

Abstract Text (1):
An apparatus and a method for constructing a multidimensional index tree which minimizes the time to access data objects and is resilient to the skewness of the data. This is achieved through successive partitioning of all given data objects by considering one level at a time starting with one partition and using a top-down approach until each final partition can fit within a leaf node. Subdividing the data objects is via a global optimization approach to minimize the area overlap and perimeter of the minimum bounding rectangles covered by each node. The current invention divides the index construction problem into two subproblems: the first one addresses the tightness of the packing (in terms of area, overlap and perimeter) using a small fan out at each index node and the other one handles the fan out issue to improve index page utilization. These two stages are referred to as binarization and compression. The binarization stage constructs a binary tree such that the entries in the leaf nodes correspond to the spatial data objects. The compression stage converts the binary tree into a tree for which all but the leaf nodes and the parent nodes of all leaf nodes have branch factors of M. In the binarization stage, a weighting or skew factor is used to achieve flexibility in determining the number of data objects to be included in each of the partitions to obtain a tree structure with desirable query performance. Thus the index tree constructed is not required to be height balanced. This provides a means to trade-off imbalance in the index tree in order to reduce the number of pages which need to be accessed in a query.

Brief Summary Text (5):
The need for efficient storage and retrieval of multidimensional objects is increasing as various applications in multimedia, digital libraries, virtual reality and information warehousing become popular. Typical two-dimensional objects include cartographic maps of countries and bands of satellite imagery. Typical 3-dimensional objects include medical images, such as MRI brain scans. These applications require the storage of data objects, some of which may be point sets while others may have non-zero measure. Such multidimensional or spatial data objects are not generally well represented by traditional structures, such as B-trees, which use a 1-dimensional ordering of key values. Multidimensional (also called spatial) data structures are well known in the art. See for example, The Design and Analysis of Spatial Data Structures, by H. Samet, Volumes 1 and 2, Addison-Wesley (1989), which is hereby incorporated by reference in its entirety.

Brief Summary Text (21):
The present invention divides the index construction or packing problem into two steps. The first (binarization) step optimizes packing density (i.e. minimizes area and overlap of the MBR covered by each index node) but uses a small fan out (say 2) at each index node. The second (compression) step optimizes fanout to improve index page utilization by increasing the fanout as nearly as possible to a predetermined value. A process having features of the present invention comprises two steps:

Detailed Description Text (14):
Those skilled in the art will appreciate that it is therefore sufficient to describe the binarization process for an arbitrary node A as depicted in FIG. 3. By way of overview, if N.sub.A .ltoreq.M, (where M is the branch factor and N.sub.A is the leaf number of node A) then node A is made a leaf node and the process stops. Otherwise, a minimum bounding rectangle (MBR) I.sub.A is considered, and a longest dimension k for this rectangle is preferably chosen. (Choosing the longest dimension favors the creation of relatively square minimum bounding

rectangles.) Each of the N.sub.A data objects 350 is then represented by their geometric centers, and they are ordered by increasing values of their kth coordinate, i.e., their position in the longest dimension. Candidate partitions (B.sub.1, B.sub.2) considered will have the property that B.sub.1 contains the first q data objects according to this order, B.sub.2 contains the remaining N.sub.A -q data objects, such that p.multidot.N.sub.A .ltoreq.q.ltoreq. (1-p).multidot.N.sub.A.

Detailed Description Text (15):
FIG. 3 shows an original node A and candidate partitions (B.sub.1, B.sub.2) which will be used in conjunction with FIG. 4a to illustrate the Binarization process of step 201. Here, the bounding rectangle is longest in the x-dimension. Assume that overlap factor, p=1/4 and that leaf number N.sub.A is 20. Thus, the leaf number condition (p.multidot.N.sub.A) requires that N.sub.B.sbsb.1 and N.sub.B.sbsb.2 must be at least 5.

Detailed Description Text (17):
In step 320, it is determined if the leaf number N.sub.A is greater than the branch factor M. If the leaf number N.sub.A >M, then in step 325 the longest dimension k of I.sub.A is determined. Otherwise, the process returns to step 310, as above. In step 330, the sweep process uses the skew factor p and o (as explained above) to determine the optimal child nodes B.sub.1 and B.sub.2 of A. In step 335, nodes B.sub.1 and B.sub.2 are added to the back of list L and the process returns to step 310.

Detailed Description Text (30):
Second, the sweep process can be performed in all dimensions, not just the longest. This may improve the solution in terms of the area and overlap, at the expense of perimeter optimization (and computational time). FIG. 8, for example, also shows a sweep process in a second dimension. This y-dimension is the shortest dimension in the example. Yet another alternative would be to use Guttman's linear or quadratic split process (as described in R-Trees: A Dynamic Index Structure for Spatial Searching, Proceedings of the ACM SIGMOD Conference (1984), by A. Guttman) suitably modified to obey the rules on partition size. (These two processes were originally proposed to handle insertions in R-trees. Because it is quadratic in the number of data objects, the latter split process may be prohibitively expensive, at least for use at high levels in the binarization process. But some combination of the linear split process at high levels of the tree and the quadratic split process at low levels may constitute a reasonable alternative strategy.)

CLAIMS:

5. A computerized method as claimed in claim 4, wherein said step of partitioning comprises the steps of:

ordering centers of the data objects according to a longest dimension in a bounding rectangle corresponding to said parent node;

evaluating all partitions according to the skew factor and for which the number of the centers of the data objects associated with the first child node is a multiple of a predetermined incremental value; and

choosing among said all partitions, one partition associated with bounding rectangles having a smallest total area subject to an overlap factor which constrains an amount of overlap of the bounding rectangles.

8. A computerized method as claimed in claim 5, comprises the steps of:

ordering centers of the data objects according to each dimension in the bounding rectangle corresponding to said parent node; and

said step of choosing among all said partitions is done in said each dimension.

18. A computer processing system as claimed in claim 17, wherein said sweep logic means comprises:

means for ordering centers of the data objects according to a longest dimension in a bounding

rectangle corresponding to said parent node;

means for evaluating all partitions according to the skew factor and for which the number of the centers of the data objects associated with the first child node is a multiple of a predetermined incremental value; and

means for choosing among said all partitions, one partition associated with bounding rectangles having a smallest total area subject to an overlap factor which constrains an amount of overlap of the bounding rectangles.

21. A computer processing system as claimed in claim 18, wherein said sweep logic comprises:

means for ordering centers of the data objects according to each dimension in the bounding rectangle corresponding to said parent node; and

means for choosing among all said partitions in said each dimension.

30. A program storage device as claimed in claim 29, wherein said step of partitioning comprises the steps of:

ordering centers of the data objects according to a longest dimension in a bounding rectangle corresponding to said parent node;

evaluating all partitions according to the skew factor and for which the number of the centers of the data objects associated with the first child node is a multiple of a predetermined incremental value; and

choosing among said all partitions, one partition associated with bounding rectangles having a smallest total area subject to an overlap factor which constrains an amount of overlap of the bounding rectangles.

33. A program storage device as claimed in claim 30, comprises the steps of:

ordering centers of the data objects according to each dimension in the bounding rectangle corresponding to said parent node; and

said step of choosing among all said partitions is done in said each dimension.

Previous Doc Next Doc Go to Doc#